

大学生平衡膳食食谱的优化设计及评价

008090

大学生平衡膳食食谱的优化设计及评价

摘要

改善大学生饮食结构不合理及不良饮食习惯的现状，提高身体健康水平，是当代大学亟待解决的问题。本文将分析大学生饮食情况，通过对男女大学生的一日食谱的调查分析，对其进行营养种类，含量测定，从而对学生一日饮食作出相应评价并依据大学食堂食谱进行优化调整。文中以蛋白质氨基酸评分最大为目标建立优化模型，用餐费用最经济为目标建立优化模型和二者兼顾建立优化模型，分别设计男生和女生的日食谱和周食谱，并对食谱进行营养评价，提供平衡膳食食谱更加有效，科学，合理的保障。在上述模型的基础上，通过对大学食谱进行稍许改良，能够有效满足大学生对营养和能量需求，提高膳食平衡，均衡营养，健康饮食。

针对问题一，首先通过对男大学生和女大学生一日三餐的食谱的所有营养成分分别进行统计，将收集到的数据绘制为图表，与标准数据相比对，然后进行对营养元素的分析、能量营养素供应统计，并且建立了氨基酸评分 $S = \frac{\sum x_{i,j} \cdot m_{r,i,j}}{\sum x_{i,j} \cdot m_{c,i,j}} \times 100\%$ 体系，接下来进行统计和分析数据。我们得出了结论，男大学生和女大学生的饮食均有不合理的部分，需要进行调整和改进。

针对问题二，首先根据所收集的数据进行限制条件的设置，然后分别根据氨基酸评分最大化 S ，用餐费用最经济 $C = \sum_{i,j} x_{i,j} \cdot C_{i,j}$ 两方面，设置日能量摄入、功能物质产能、三餐能量分配、非产能主要营养素、食物种类等限制条件，建立线性规划模型，最后根据模型使用 python 进行问题的模拟，分别是氨基酸评分最大、最经济以及兼顾二者 ($\omega_1 S - \omega_2 C$) 的模拟，每种食谱均有其利弊。

针对问题三，我们在问题二的基础上，增加了周饮食限制，包括但不限于食物种类需要多于 25 种，建立了周饮食模型，即多目标规划。对数据进行多次测试，总结最优周食谱方案，最后得出结论，在允许的情况下，我们需要兼顾营养和经济，才能在大学生活中健康成长。

针对问题四，我们综合前三个问题得到的数据，查询倡议书格式，书写倡议书，力求更多的大学生健康饮食，健康成长。

本文的特色是将数据分析整理和多目标规划相结合，将多目标问题转换为单目标问题，使求解更加渐变易懂，逻辑清晰。

关键词：线性规划；多目标规划；大学生饮食

一、问题的重述

1.1 问题背景

大学是我们学习新知与重塑体魄的全新时段。在大学这个黄金时期，我们面临课业压力的脑部劳力和身体方面的硬性需求，对能量和营养有着更加严苛的要求。这个阶段的年轻人，每日学习知识对脑部的锻炼日益强化，身体蓬勃发展，更需进一步的营养补充和能量补给，从而为大学美好充实生活增添厚实的保障。

近年来，大学校园里饮食结构失衡与不良饮食习惯的问题尤为突出。诸如忽略早餐，或早餐敷衍了事，过度依赖外卖快餐，甚至有学生为减少皮下脂肪而控制进食，导致营养不良。在这样的背景下，大学阶段学会营养知识，养成良好饮食习惯，显得至关重要。它不仅有助于个体生长发育，更是保障身心健康的关键所在。

当前，大学生营养结构不合理及不良饮食习惯问题亟待解决，对此我们对男女大学生的一日三餐食物摄入情况进行记录，并由此测定其膳食食谱的营养成分推断饮食的科学合理性。通过了解该同学每日膳食食谱中一日三餐摄入营养的主要成分，定量测定两份食谱的摄入营养。针对高校学生食堂一日三餐主要食物信息统计表，我们对两份膳食食谱进行合理调整改进。

1.2 问题描述

1. 根据平衡膳食基本准则对两份美食佳作进行全面膳食营养评估并对两份美食进行细腻的调整优化，再次进行全面膳食营养评估。
2. 基于该高校学生食堂一日三餐主要食物信息统计表的日平衡膳食食谱的优化设计和周平衡膳食食谱优化设计，建立线性回归模型，设计男生和女生的日食谱，然后分别对食谱进行膳食营养评价。
3. 基于问题2建立多目标模型，设计更多的限制条件，设计出男生和女生的周食谱，并进行比较分析
4. 针对大学生饮食结构及习惯，对大学生健康饮食，平衡膳食发出倡议

二、问题的分析

2.1 问题一的分析

针对问题一，我们根据已有提供的数据，运用 excel 查出各食物种类营养成分含量，统计出男大学生女大学生一日三餐食谱中各食物种类营养成分含量及能量占比，并做出对应的图表，从而与标准数据作对比，对两份食谱进行营养评价及优化调整。

2.2 问题二的分析

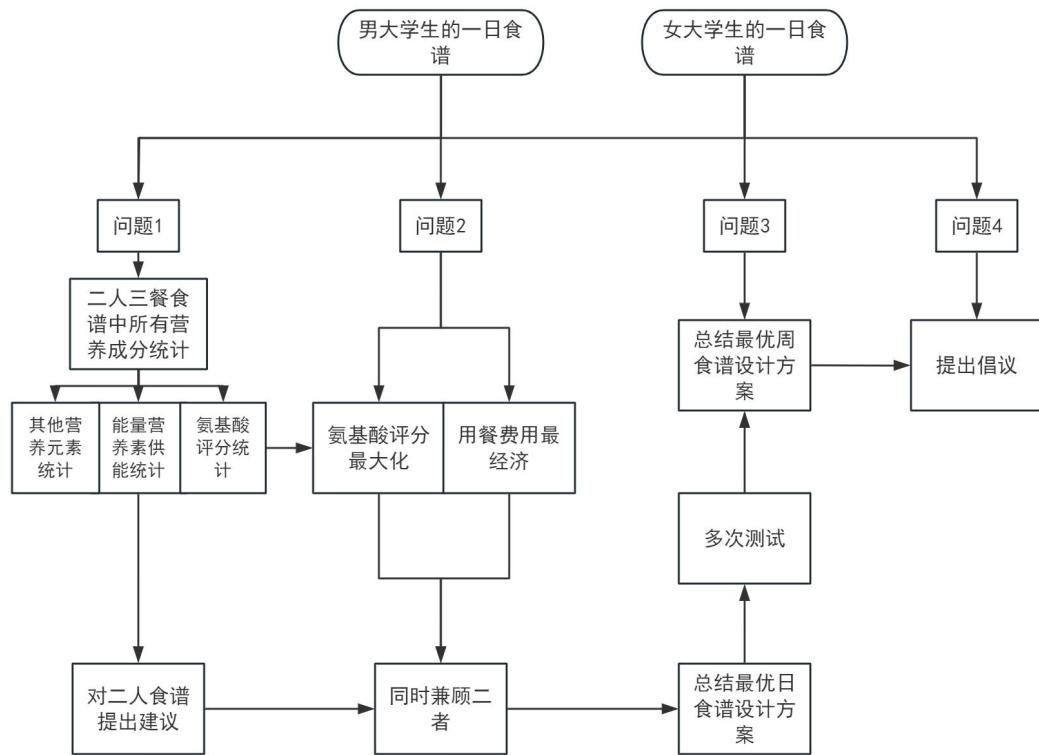
针对问题二，我们使用蛋白质氨基酸评分和用餐费用最经济，和兼顾二者三种种建立优化模型，运用线性回归，单目标规划，以及多目标规划进行数据分析。

2.3 问题三的分析

针对问题三，我们考虑更多的限制条件，建立周饮食模型，多目标规划，将问题二放大化，提供更多的食物种类，给出最优周食谱方案，给作参考。

2.4 问题四的分析

针对大学生的生活习惯以及经济水平，综合问题一二三，给出倡议，呼吁更多大学生进行平衡膳食食谱的优化。



三、基本假设

1. 不考虑其他外界因素（如食堂的卫生条件、食品的质量差异）对学生的进食影响
2. 假设每份餐各成分含量固定
3. 假设学生饮食时间固定且除进餐时间外没有其他能量摄入

四、定义符号说明

序号	符号	符号说明
1	S	氨基酸蛋白质评分
2	E	能量
3	ϵ	差距率
4	C	用餐费用

五、问题的分析

5.1 问题一：膳食食谱的营养分析及调整

5.1.1 问题一的分析

膳食平衡是指在日常生活饮食中，需保证食物种类的多样性，摄入量的适度性，营养素比例的合理性，以及营养摄入与消耗之间的相对平衡。当前，我国营养与健康方面仍存在一定问题，如营养素摄入不足，膳食不平衡，营养素比例结构不合理等，这些问题导致许多人处于亚健康状态。为改善这一状况，我们需要关注饮食平衡的合理性，因为膳食平衡是实现健康的基础标准。[4]

5.1.2 问题一的解决

根据附件1，附件2中的男女大学生一日食谱调查，可初步得出：

男大学生：

白菜、稻米、豆油、干豆腐、海带、鸡蛋、木耳、茄子、青椒、酸菜、土豆、五花猪肉、小麦粉、小米、油菜、玉米面、炸鸡块、芝麻油、猪肉。

去除相同成分后共19种；

女大学生：

白菜、稻米、豆油、黄豆、火腿肠、鸡蛋、鸡肉、苹果、葡萄、茄汁沙丁鱼、蒜苔、香菇、小麦粉、油菜、猪肉。

去除相同成分后共15种；

依据平衡膳食的基本准则：二者均满足每日摄取食物大于12种的要求；

将其分为五大类表格如下：

	类别	男	女
1	谷、薯类	稻米、小麦粉、小米、玉米面	稻米、小麦粉
2	蔬菜、菌藻、水果类	白菜、木耳、茄子、青椒、酸菜、土豆、油菜、海带	白菜、苹果、葡萄、油菜、香菇、蒜苔
3	畜、禽、鱼、蛋类及制品	五花猪肉、炸鸡块、鸡蛋、猪肉	火腿肠、鸡蛋、鸡肉、茄汁沙丁鱼、猪肉
4	奶、干豆、坚果、种子类及制品	干豆腐	黄豆
5	植物油类	豆油、芝麻油	豆油

可见、同时满足摄取的食物包含五大类别。

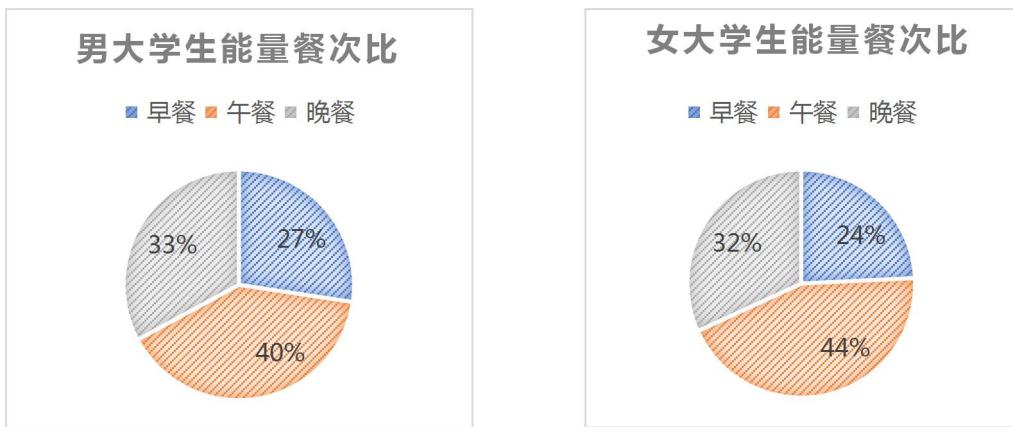
食谱中各种营养素含量见附录。

附录中整理出：

该男大学生该日能量摄入 $2837.19kcal$;

该女大学生该日能量摄入 $1228.43kcal$.

且三餐能量分配占总能量的百分比如下图



与参考值差距：

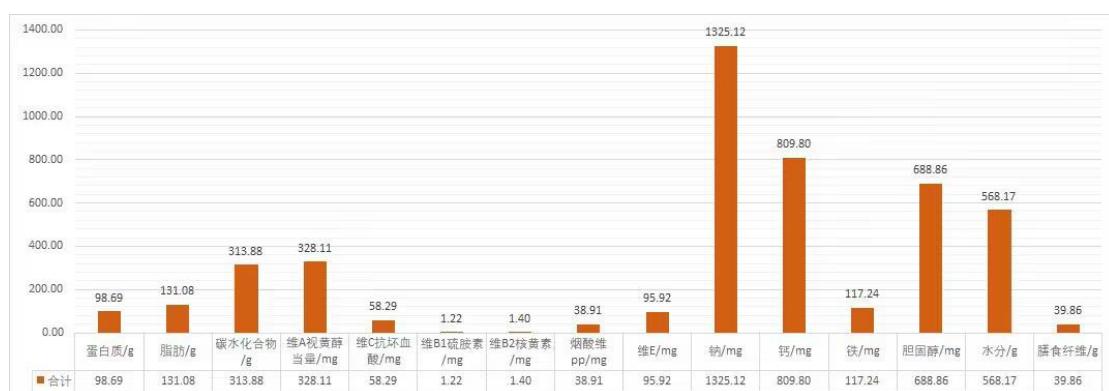
	早餐	中餐	晚餐
参考值	30%	30%—40%	30%—40%
男大学生	27%	40%	33%
差距	↓	—	—
女大学生	24%	44%	32%
差距	↓	↑	—

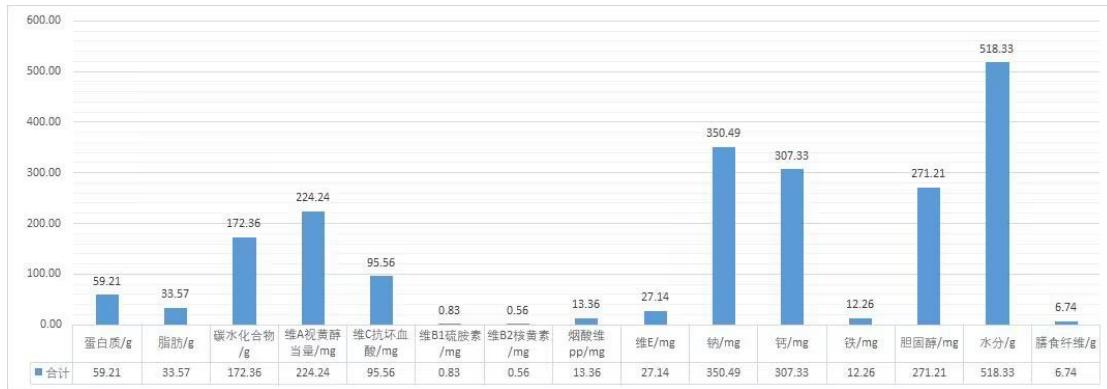
大学生每日的能量摄入目标为：女生 1900kcal/d，男生 2400kcal/d.

男大学生每日摄入能量超过目标值接近 2 成，能量摄入量过多，能量餐次与参考值接近，较为合理，整体需要减少能量的摄入，保持三餐合理的能量摄入比；

女大学生一日摄入能量低于目标值约 35%，摄入量明显不足，能量餐次比也与参考值有较大偏差，总体需要摄入更多的能量，将能量餐次比调整为 3:3.5:3.5 范围内，存在较大优化改进空间。

下图分别为男大学生、女大学生一日三餐合计的成分情况





每日膳食非产能主要营养素参考摄入量：

营养素	钙 (mg·d ⁻¹)	铁 (mg·d ⁻¹)	维生素 A (μg·d ⁻¹)	维生素 B ₁ /硫胺素 (mg·d ⁻¹)	维生素 B ₂ /核黄素 (mg·d ⁻¹)	维生素 C (mg·d ⁻¹)
男生	800	12	800	1.4	1.4	100
女生	800	20	700	1.2	1.2	100

维生素与矿物质是维持生物体生命活动不可或缺的物质。大部分矿物质和部分维生素无法在机体内部合成，且不能在组织内大量储存，因此，必须通过食物摄入。由于饮食行为不佳，维生素和矿物质的缺乏现象在人群中较为广泛。根据膳食调查结果，男女学生在维生素A、维生素B₁、维生素B₂、钙和锌的摄入上都存在不足，而烟酸的摄入量则达到要求。具体来说，男生在维生素A、维生素C和钙的摄入方面严重不足，女生则在维生素A、维生素B₂、钙和铁的摄入上严重不足。造成维生素C不足的主要原因在于大学生对果蔬类食物的摄入量不足。钙的缺乏则与奶及奶制品的摄入不足有关。值得注意的是，男生的铁摄入量高于女生，这可能与男生摄入较多肉类有关，因为肉类中的铁含量较高且易于吸收。[3]

由上述图表可知：

该男大学生一日摄入的钙含量符合参考量，铁含量超标，维生素A摄入量不足，维生素B₁, B₂摄入量达到期望值，维生素C摄入量不足；

此女大学生三餐摄入的钙，铁含量均不足，维生素A摄入量不足，维生素B₁, B₂摄入量均不达标，维生素C摄入量符合预期值。

两份食谱每日膳食非产能主要营养素摄入量均存在明显不足，男女大学生摄入营养符合预期值较少，营养摄入量不充足或存在摄入超标，三餐营养摄入，膳食平衡问题凸显。

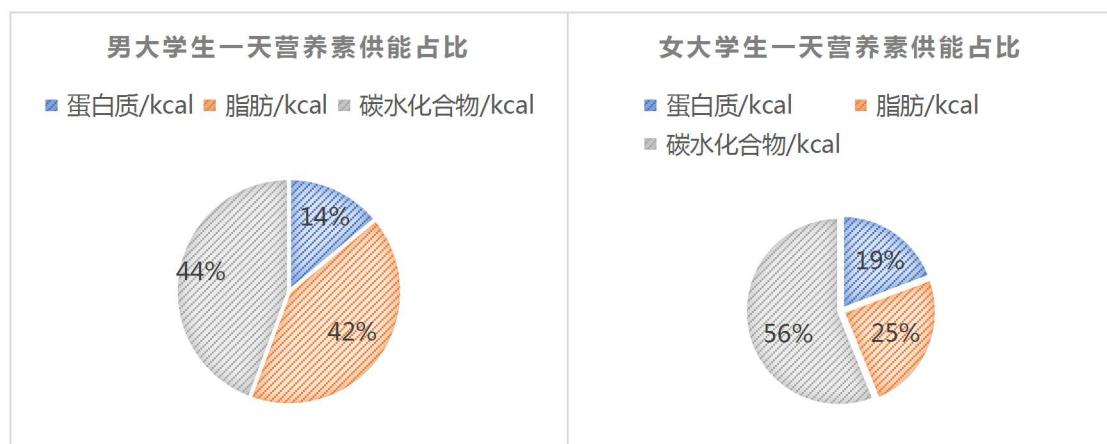
将碳水化合物，脂肪，蛋白质转化成能量。

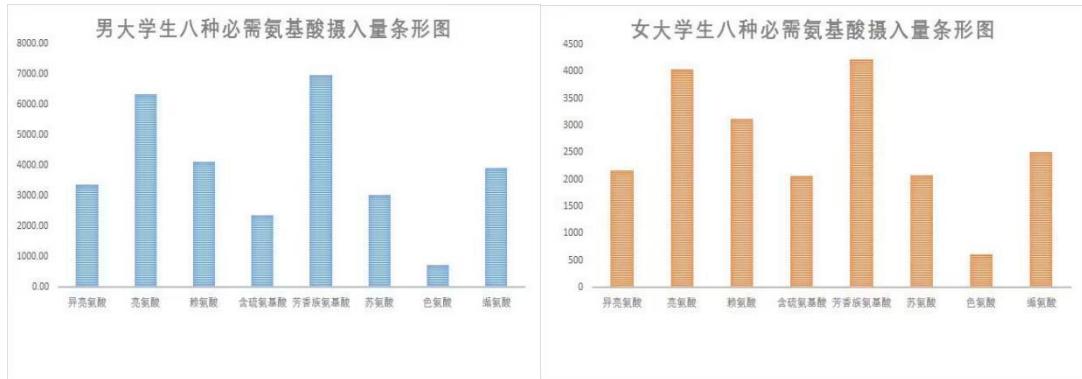
男	能量/kcal	蛋白质/ kcal	脂肪/ kcal	碳水化合物/ kcal
早餐	782.17	80.6	349.21	352.36
午餐	1134.94	139.6	451.82	543.52
晚餐	939.87	174.55	405.69	359.62
总计	2856.98	394.75	1206.73	1255.51
女	能量/kcal	蛋白质/ kcal	脂肪/ kcal	碳水化合物/ kcal
早餐	308.63	65.68	82.31	160.64
午餐	559.26	75.3	193.67	290.3
晚餐	405.44	95.86	71.08	238.5
总计	1273.33	236.84	347.05	689.44

根据《中国居民膳食指南》：大学生群体每日宏量营养素供能占总能量的百分比参考值分别为：蛋白质 10%-15%、脂肪 20%-30%、碳水化合物 50%-65%。

该男大学生一日摄入营养素供能占能量比中蛋白质处于合理区间，脂肪的占比明显超出参考范围，碳水化合物的占比略低于预期范围。由此可见，这位男大学生一日摄入的蛋白质含量较为合理，脂肪摄入量过多，碳水化合物摄入量相对较少，鉴于该男大学生三餐食谱总摄入量超出目标值，因此在减少能量摄入中要注重少吃富含脂肪的食物，该食谱的能量占比结构不合理，应着重减少脂肪摄入，适当减少碳水化合物摄入；

该女大学生一日摄入营养素供能占能量比中蛋白质摄入量占比偏大，脂肪摄入量占比合理，碳水化合物比例在预期范围内。由上述图表，这位女大学生一日摄入的蛋白质较多，脂肪，碳水化合物的占比较为合理；因为该女大学生能量总摄入量不足，故应该在补充营养的过程中适当多食用碳水化合物占比高，脂肪含量恰当的食物，对富含蛋白质的食物要加以控制。





评分	女	男
早餐	156.10%	59.85%
午餐	54.30%	44.69%
晚餐	42.66%	86.40%
合计	63.75%	60.43%

混合食物蛋白质氨基酸评分规则

$S < 60$	$60 \leq S < 80$	$80 \leq S < 90$	$S \geq 90$
不合理	不够合理	比较合理	合理

其中 S 为某餐的氨基酸评分.

在对男女大学生三餐食谱中混合食物蛋白质氨基酸评分中。

女大学生早餐合理，午餐，晚餐均不合理，该日评分为 63.75%，不够合理；

男大学生早餐，午餐不合理，晚餐比较合理；该日评分为 60.43%，不够合理；

二者三餐混合食物蛋白质氨基酸评分存在明显差异，且总评分均较低，不够合理。

	能量/kcal	蛋白质/kcal	脂肪/kcal	碳水化合物/kcal
男实际能量供应	2909.73	394.75	1179.75	1255.51
男标准能量供应	2400.00	300.00	600.00	1380.00
差距率	+21.23%	+31.58%	+96.63%	-9.02%
女实际能量供应	1241.86	236.84	302.10	689.44
女标准能量供应	1900.00	237.50	475.00	1092.50
差距率	-34.64%	-0.28%	-36.40%	-36.89%

差距率

$$\epsilon = \frac{E_r - E_c}{E_c} \times 100\%$$

其中 E_r 、 E_c 分别为实际能量供应和标准能量供应。

根据上述测定结果，两份食谱符合食物种类大于 12 种，五大类食物齐全的基本要求；在食谱中，男大学生能量摄入量偏大，餐次比更加符合参考标准；女大学生能量摄入量

偏少，餐次比与标准有明显差距。两份食谱的非产能营养素含量与标准值有较大差异；男大学生食谱中碳水化合物、脂肪、蛋白质供能占总能量的百分比脂肪比例较高，碳水化合物比例较低，女大学生蛋白质占比偏高，碳水化合物占比偏低；男女大学生每餐混合食物的蛋白质氨基酸评分差异较大，均为不够合理。

对此，该男大学生要适当减少每日食物的摄入，保持在每日 2400kcal 左右，女大学生要合理增加食物摄取，达到每日 1900kcal，同时将餐次比进行调整至合理水平。二人要注重对非产能营养素的摄入，男大学生要多补充含维生素 A，维生素 C 的食品，减少含铁食品的食用，女大学生要多补充含钙，铁元素，维生素 A 维生素 B1 B2 的食品。男大学生在减少每日饮食食物中要重点减少脂肪摄入，女大学生要增加脂肪和碳水化合物的补充，让产能营养素供能在合理水平。男女大学生要多补充食品中蛋白质氨基酸评分较高的食品，实现蛋白质的互补，让每日的评分保持在比较合理及以上水平。

下表为改进过后的男大学生和女大学生的食谱

调整后男大学生的1日食谱					调整后女大学生的1日食谱				
早餐					早餐				
食物名称	主要成分	食物编码	可食部(克/份)	食用份数	食物名称	主要成分	食物编码	可食部(克/份)	食用份数
小米粥	小米	015101	15	1	豆浆	黄豆	031101	10	1
油条	小麦粉	011201x	50	1		小麦粉	011201x	50	
	豆油	192004	10			鸡肉	091112	40	1
煎鸡蛋	鸡蛋	111101x	50	1		豆油	192004	5	
	豆油	192004	10			小麦粉	011201x	50	1
拌海带丝	海带	052004	100	1		豆油	192004	10	
	芝麻油	192017	2						
午餐					午餐				
食物名称	主要成分	食物编码	可食部(克/份)	食用份数	食物名称	主要成分	食物编码	可食部(克/份)	食用份数
大米饭	稻米	012001x	25	3	鸡蛋饼	小麦粉	011201x	25	
拌木耳	木耳	051014	100	1		鸡蛋	111101x	20	
	芝麻油	192017	2			火腿肠	081409	20	1
地三鲜	茄子	043101x	80			豆油	192004	5	
	土豆	021101	80	1		小麦粉	011201x	50	
	青椒	043123	10			猪肉	081111x	20	
	豆油	192004	10			白菜	045101x	40	1
红烧肉	五花猪肉	081108	50			豆油	192004	10	
	干豆腐	031509	50	1		葡萄	063101x	100	1
	豆油	192004	10						
晚餐					晚餐				
食物名称	主要成分	食物编码	可食部(克/份)	食用份数	食物名称	主要成分	食物编码	可食部(克/份)	食用份数
砂锅面	玉米面	013109	80		大米饭	稻米	012001x	25	2
	白菜	045101x	20			油菜	045125	100	
	油菜	045125	20	1		香菇炒油菜	香菇	051019	20
	干豆腐	031509	10			豆油	192004	5	
	豆油	192004	10			蒜台	044106	100	
包子	小麦粉	011201x	25			炒肉蒜台	猪肉	081111x	30
	猪肉	081111x	15	1			豆油	192004	5
	酸菜	045119	20			茄汁沙丁鱼	茄汁沙丁鱼	121417	100
	豆油	192004	5			苹果	苹果	061101x	100
炒牛肉	牛肉	082105	100	0.5					1
	青椒	043123	20						1/2
	胡萝卜	041201	10						
	豆油	192004	10						

其中改进过后的氨基酸评分分别为 89 和 86.

5.2 问题二：基于附件 3，对附件 1、附件 2 两份食谱进行较少的改进调整，并且再做出全面的膳食营养评价

5.2.1 问题二的分析

我们使用蛋白质氨基酸评分和用餐费用最经济，和兼顾二者三种种建立优化模型，运用线性回归，单目标规划，以及多目标规划进行数据分析。

5.2.2 模型的建立

5.2.2.1 以蛋白质氨基酸评分最大为目标建立优化模型

有研究显示，在校大学生普遍存在维生素和蛋白质摄入量不足、脂肪热比过高及偏食等问题。学校应引导学生注重蛋白质、碳水化合物、脂肪、维生素和矿物质的科学搭配，提供足够的能量和营养素，确保营养均衡，保证每餐的摄入量都在可控范围之内。

以蛋白质氨基酸评分为目标函数，建立模型. [2]

决策变量： 学生在第 j 餐中选择的第 i 种食物质量(g) : $x_{i,j}$.

目标函数： 蛋白质氨基酸评分.

$$maxS = \frac{\sum x_{i,j} \cdot m_{r,i,j}}{\sum x_{i,j} \cdot m_{c,i,j}} \times 100\%$$

其中 S 为蛋白质氨基酸评分; $m_{r,i,j}$ 、 $m_{c,i,j}$ 分别为第 j 餐中选择的第 i 种食物中必须氨基酸的含量和参考蛋白质氨基酸含量.

约束条件：

① 日能量摄入；

$$0.9 \times E_t \leq \sum_{i,j} x_{i,j} \cdot E_i \leq 1.1 \times E_t$$

其中 E_i 为第 j 餐中选择的第 i 种食物中能量含量， E_t 为标准日能量摄入

$$E_t = \begin{cases} 2400, & \text{男生} \\ 1900, & \text{女生} \end{cases}$$

② 蛋白质(代号 1), 脂肪 (代号 2) , 碳水化合物(代号 3) 产能百分比;

$$dn_k \times E_t \leq \sum_{i,j} x_{i,j} \cdot e_{i,k} \leq up_k \times E_t$$

其中 k 可取 1, 2, 3. $e_{i,k}$ 是第 i 种食物中 k 物质的含量， dn_k , up_k 分别为 k 占总能量百分比上下界.

$$dn_k = \begin{cases} 10\%, & k = 1 \\ 20\%, & k = 2 \\ 50\%, & k = 3 \end{cases} \quad up_k = \begin{cases} 15\%, & k = 1 \\ 30\%, & k = 2 \\ 65\%, & k = 3 \end{cases}$$

③ 三餐能量分配占总能量的百分比;

$$d_j \times E_t \leq \sum_i x_{i,j} \cdot E_i \leq u_j \times E_t$$

固定 j 分别为 1, 2, 3.

其中 d_j, u_j 为三餐餐次比上下界

$$d_j = \begin{cases} 25\%, j = 1 \\ 30\%, j = 2 \\ 30\%, j = 3 \end{cases} \quad u_j = \begin{cases} 35\%, j = 1 \\ 40\%, j = 2 \\ 40\%, j = 3 \end{cases}$$

④ 非产能主要营养素限制:

$$0.9 \times h_k \leq \sum_{i,j,k} x_{i,j} \cdot h_{i,k} \leq 1.1 \times h_k$$

其中 h_k 为营养素参考值见下表 (k 为代号)

营养素	钙 (mg. d-1)	铁 (mg. d-1)	维生素 A (μg. d-1)	维生素 B1/硫 胺素 (mg. d-1)	维生素 B2/核 黄素 (mg. d-1)	维生素 C (mg. d-1)
男生	800	12	800	1.4	1.4	100
女生	800	20	700	1.2	1.2	100

其中 $h_{i,k}$ 为第 i 种食物中 k 物质的含量,

⑤ 食物种类大于 12 种

$$i > 12$$

5.2.2.2 以用餐费用最经济为目标建立优化模型

经调查结果显示, 餐食的价格, 性价比是在校大学生在饮食时的重要考虑因素。学校在提供营养丰富, 口感美味的食品之余, 也要考虑餐品价格, 确保部分学生在经济水平有限之时, 保证其每餐的价格在可控范围之内。

决策变量: 学生在第 j 餐中选择的第 i 种食物质量(g): $x_{i,j}$.

目标函数: 用餐费用.

$$\min C = \sum_{i,j} x_{i,j} \cdot C_{i,j}$$

其中 C 为第 j 餐中选择的第 i 种食物的价格.

满足约束条件①②③④⑤.

5.2.2.3 兼顾蛋白质氨基酸评分及经济性, 建立优化模型

如果兼顾蛋白质氨基酸评分和经济性, 需要将上述两个目标函数兼顾.

决策变量: 学生在第 j 餐中选择的第 i 种食物质量(g): $x_{i,j}$.

目标函数:

$$\max(\omega_1 S - \omega_2 C)$$

其中 ω_1, ω_2 分别为蛋白质氨基酸评分和经济性的权重. ($\omega_1 = \omega_2 = 0.5$)

满足约束条件①②③④⑤.

5.2.2.4 基于 1-3 模型对日平衡膳食食谱进行比较分析

```

IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938 64 bit (^
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> = RESTART: C:\Users\shadd\Desktop\2.py
男大学生蛋白质氨基酸评分最高:
1 2 3
牛奶 大米饭 鸡排面
大米粥 红烧牛肉面 水饺
煮鸡蛋 拌菠菜 拌干豆腐
苹果 橙子 葡萄
S=97
C=35

女大学生蛋白质氨基酸评分最高:
1 2 3
牛奶 大米饭 鸡排面
小米粥 红烧牛肉面 水饺
煮鸡蛋 拌菠菜 拌干豆腐
橙子 苹果 葡萄
S=98
C=34

男大学生最经济:
1 2 3
豆浆 包子 鸡蛋饼
馒头 红烧牛肉面 水饺
小米粥 拌海带丝 拌土豆丝
香蕉 西瓜 柚子
S=89
C=24

女大学生最经济:
1 2 3
豆浆 包子 鸡蛋饼
馒头 红烧牛肉面 水饺
小米粥 拌海带丝 拌土豆丝
香蕉 西瓜 柚子
S=82
C=20

```

下图一日菜谱为以蛋白质氨基酸评分最大: (左男右女, 1,2,3 为餐次代号)

1	2	3	1	2	3
牛奶	大米饭	鸡排面	牛奶	大米饭	鸡排面
大米粥	红烧牛肉面	水饺	小米粥	红烧牛肉面	水饺
煮鸡蛋	拌菠菜	拌干豆腐	煮鸡蛋	拌菠菜	拌干豆腐
苹果	橙子	葡萄	橙子	苹果	葡萄
S=97			S=98		
C=35			C=34		

以用餐费用最经济:

1	2	3	1	2	3
豆浆	包子	鸡蛋饼	豆浆	包子	鸡蛋饼
馒头	红烧牛肉面	水饺	馒头	红烧牛肉面	水饺
小米粥	拌海带丝	拌土豆丝	小米粥	拌海带丝	拌土豆丝
香蕉	西瓜	柚子	香蕉	西瓜	柚子
S=89			S=82		
C=24			C=20		

兼顾蛋白质氨基酸评分及经济性：

1	2	3	1	2	3
牛奶	大米饭	鸡排面	牛奶	大米饭	鸡排面
大米粥	红烧牛肉面	水饺	小米粥	红烧牛肉面	水饺
煮鸡蛋	拌菠菜	拌干豆腐	煮鸡蛋	拌菠菜	拌干豆腐
苹果	橙子	葡萄	橙子	苹果	葡萄
S=93			S=92		
C=27			S=93		

```

*IDLE Shell 3.12.3*
File Edit Shell Debug Options Window Help
目标函数: [ 1.5 1.5 1.5 0.5 0.5 0.5 1. 1. 1.5 0. 1. 1.5 0. 1.
0.5 0. 6. 0. 0. 6. 0. 0. 6. 0. 0. 0. 0. 0. 2.
0. 0. 0. 2. 0. 0. 0. 3. 0. 0. 0. 2.5 0. 0.
0. 2. 0. 0. 8. 0. 0. 0. 2. 0. 0. 0. 0. 0. 2.
2. 0. 2. 0. 2. 0. 2. 0. 2. 0. 3. 0. 0. 0. 1.
1. 1. 0.5 1. 1. 1.5 0.5 0. 6. 0. 0. 6. 0. 0. 0.
6. 0. 0. 0. 8. 0. 0. 0. 8. 0. 0. 0. 0. 0. 0.
2. 0. 0. 0. 2.5 0. 0. 0. 3. 0. 0. 0. 0. 2.5 0.
0. 0. 2. 0. 0. 8. 0. 0. 0. 10. 0. 0. 0. 0. 2.
0. 0. 0. 2. 0. 0. 0. 2. 0. 0. 0. 3. 0. 0. 0.
2. 0. 0. 2. 0. 2. 0. 2. 0. 3. 0. 0. 3. 0. 0. 0.
0. 3. 0. 0. 5. 0. 0. 0. 5. 0. 0. 0. 3. 0. 0. 0.
3. 0. 0. 3. 0. 0. 3. 0. 0. 0. 3. 0. 0. 0. 4.
0. 0. 4. 0. 0. 4. 0. 0. 0. 5. 0. 0. 5. 0. 0.
0. 5. 0. 0. 5. 0. 0. 5. 0. 0. 0. 5. 0. 0. 0.
8. 0. 0. 0. 8. 0. 0. 0. 8. 0. 0. 8. 0. 0. 0.
8. 0. 0. 0. 0. 7. 10. 0. 0. 0. 7. 7. 0. 0. 7.
0. 0. 0. 1. 1. 1. 1. 1. 0.5 1. 1. 1.5 0.5 6.
0. 0. 6. 0. 0. 6. 0. 0. 0. 0. 2. 0. 0. 0. 0.
3. 0. 0. 0. 2.5 0. 0. 0. 8. 0. 0. 0. 8. 0. 0.
0. 0. 2. 0. 0. 0. 2. 0. 0. 0. 2. 0. 2. 0. 0.
2. 0. 2. 0. 3. 0. 0. 3. 0. 0. 0. 5. 0. 0. 0.
0. 5. 0. 0. 3. 0. 0. 3. 0. 0. 3. 0. 0. 0. 3.
0. 0. 0. 3. 0. 0. 4. 0. 0. 4. 0. 0. 5. 0. 0.
0. 5. 0. 0. 5. 0. 0. 5. 0. 0. 5. 0. 0. 0. 5.
0. 0. 0. 8. 0. 0. 0. 8. 0. 0. 8. 0. 0. 0. 0.
0. 7. 10. 0. 0. 7. 0. 7. 0. 7. 0. 0. 0. 1.
1. 1. 1. 1. ]
约束矩阵 (不等式): Squeezed text (327 lines).
约束向量 (不等式): [65, 2100, 70, 280]
约束矩阵 (等式): Squeezed text (327 lines).
约束向量 (等式): [45, 1700, 50, 220]
变量边界: Squeezed text (53 lines).

```

5.3 问题三：基于附件3的周平衡膳食食谱的优化设计

根据研究表明，大学生在参考制定每日膳食平衡食谱时，往往因食谱单一，每日严格的营养摄入量等因素而难以维持。为了保障大学生每周饮食健康，养成良好膳食平衡习惯，本文以每日膳食平衡食谱模型为基础，对每周食谱进行合理搭配，在保障在校大学生每日饮食结构合理，餐次比达到目标，摄入营养充分的同时，满足多种食材的搭配，提供更加丰富的食物选择，实现周平衡膳食食谱的制定。

决策变量：学生在第 k 天的第 j 餐中选择的第 i 种食物质量(g): $x_{i,j,k}$. (其中 $k=1,2,\dots,7$)

第 k 天的蛋白质氨基酸评分：

$$S_k = \frac{\sum x_{i,j,k} \cdot m_{r,i,j}}{\sum x_{i,j,k} \cdot m_{c,i,j}} \times 100\%$$

第 k 天的用餐费用：

$$C_k = \sum_{i,j} x_{i,j,k} \cdot C_{i,j}$$

以蛋白质氨基酸评分最大为目标建立优化模型

目标函数:

$$\sum_{k=1}^7 S_k$$

以用餐费用最经济为目标建立优化模型

目标函数:

$$\sum_{k=1}^7 C_k$$

兼顾蛋白质氨基酸评分及经济性，建立优化模型

目标函数:

$$\omega_1 \sum_{k=1}^7 S_k - \omega_2 \sum_{k=1}^7 C_k$$

同时满足约束条件①②③④⑤.

以及周食谱评价要求大于 25 种:

$$i > 25$$

周食谱见附录 1

该周食谱兼顾了蛋白质氨基酸评分与经济性，食品评分均在比较合理及以上水平。在保持每天食品种类丰富之余保证了每周菜品种类达到 25 种以上的要求。每日能量摄入量，餐次比，碳水化合物，蛋白质，脂肪供能占比均在合理阈值，对非产能主要营养素钙、铁、锌、维生素 A、维生素 B₁、维生素 B₂、维生素 C 的摄入量在合理范围，为大学生提供了充足的营养。两份食谱中，男大学生在保障日常供能与耗能合理同时，改变了三餐中供能占比，减少了脂肪的摄入比例，对矿物质元素和维生素类的补充更加完善。女大学生则增加了每日的食物摄入，提高了每餐食品蛋白质氨基酸评分。二者周平衡膳食食谱对调查的每日食谱的优化方向有所差异，改良食谱不仅丰富了食品的类别，合理了能量的摄取，为大学生平衡膳食提供了坚实保障，引导大学生养成良好的饮食习惯，均衡营养，健康生活。

5.4 问题四：针对大学生饮食结构及习惯，写一份健康饮食、平衡膳食的倡议书

大学生健康饮食和平衡膳食倡议书

亲爱的大学生朋友们：

你们好！

在繁忙的学业和丰富的校园生活中，健康的饮食习惯往往被忽视。然而，合理饮食不仅关系到我们的身体健康，还直接影响到学习效率和生活质量。为此，我们特此倡议大家树立健康饮食和平衡膳食的观念，并在日常生活中加以践行。

一、均衡饮食，保证营养摄入

多样化食物选择：尽量选择多种食物，每日摄入食物种类不少于 12 种，一周摄取的食物要在 25 种以上，每天摄入的食物要包括五大类别：谷、薯类；蔬菜、菌藻、水果类；畜、禽、鱼、蛋类及制品；奶、干豆、坚果、种子类及制品；植物油类。确保人体摄入适量的蛋白质、脂肪、碳水化合物等能量营养素，同时，也要保证维生素、矿物质、膳食纤维等非能量营养素的摄入。只有营养素摄入均衡，才能维持人体的健康。大学生每日的能量摄入目标为：女生 1900kcal/d，男生 2400kcal/d，每日摄入合理充足的食物保障身体的能量补给；同时我们要合理安排摄入的蛋白质，碳水化合物，脂肪三种宏量营养素，产能营养素占总能量百分比应尽量满足蛋白质 10%-15%、脂肪 20%-30%、碳水化合物 50%-65%，确保自己能够有合理的能量补充。

适量摄入蛋白质：蛋白质是维持身体正常功能和肌肉健康的重要成分。可以从瘦肉、鸡蛋、豆制品、乳制品等中获取优质蛋白质。针对不同食物蛋白质的营养特点，进行氨基酸评分，每餐对蛋白质氨基酸的摄入量要达到比较合理及以上水平。

足量摄入碳水化合物：碳水化合物是主要的能量来源，应优先选择全谷物、杂粮等富含纤维的食物，减少精制糖和高糖食品的摄入。

适量摄入健康脂肪：优先选择不饱和脂肪，如坚果、鱼类、橄榄油等，减少饱和脂肪和反式脂肪的摄入。我们应该摒弃对脂肪固有的消极观念，脂肪为我们学生每天的生活提供能量，还保障了我们身体各项机能，科学适度的摄入脂肪对我们的身心健康有着重要作用。

增加蔬菜水果的摄入：每日应摄入足够的蔬菜和水果，保证维生素和矿物质的充足供应，并帮助消化和排毒。合理摄入非产能营养素，及时补充含铁，钙，锌等矿物质元素和维生素类的食品，矿物质元素和维生素类虽然不直接参与身体的能量供应，部分在人体中的含量极其微小，但它们对我们的生理功能和人体生态系统平衡起到了调节作用，平衡膳食中对非产能营养素的摄入有利于大学生健康。

二、科学安排三餐，养成良好习惯

三餐能量分配比要进行合理控制，科学的餐次比为人体每日不同时段提供了充足和适当的营养供给，餐食间隔相对的合理，能够较好地促进人体对食物的吸收与消耗，保证能量摄入与消耗平衡，避免出现体力不足，同时，三餐合理的能量分配对保持肠胃健康，稳定血糖也有着重要作用，能够有效预防消耗系统疾病。每日餐次比要争取达到早餐 30%，中餐、晚餐各 30%—40%

定时定量，养成规律的饮食习惯，三餐按时进食，避免暴饮暴食和长时间饥饿。

早餐要吃好：早餐是一天中最重要的一餐，应保证摄入充足的营养，避免空腹上课或工作。

午餐要吃饱：午餐应摄入足够的能量和营养，帮助下午保持良好的精神状态和学习效率。

晚餐要吃少：晚餐应适量，避免过量摄入高热量、高脂肪食物，影响消化和睡眠质量。

饮食要兼顾营养与经济，在注重食物的口感时关注食品的营养健康，减少攀比，不暴饮暴食，铺张浪费，也不应过少饮食，导致营养不良，能量摄入不足。

三、减少不健康饮食，避免不良习惯

少吃高糖高脂食品：减少饮用含糖饮料和吃高糖、高脂肪的零食，如薯片、巧克力等。

减少外卖频率：尽量选择健康的食物和烹饪方式，减少油炸食品和高盐食物的摄入。

控制酒精摄入：避免过量饮酒，减少对身体的伤害。

对每日饮食的改善要循序渐进，逐步改善饮食习惯，做到平衡膳食，营养均衡，合理健康。

四、注意饮食安全，避免疾病风险

一、保证食物新鲜：选择新鲜、优质的食材，避免食用过期或变质的食物。

二、注意烹饪卫生：保持厨房和餐具的清洁，生熟食物分开处理，防止食物交叉污染。

三、适量饮水：每天应摄入足够的水分，避免饮用过多含糖饮料或酒精饮品。

[1]优化大学生的膳食结构，满足其身体所需的营养元素，可以提高他们的体能、抵抗力及免疫力，使他们拥有健康的体魄，从而有利于提高学习效率。健康饮食可以改善大学生的身体状况，保持良好的健康水平，通过食物补充达到预防疾病的效果。有益于大学生心理健康。部分食物具有调节情绪的特性，面对巨大的学习与就业压力，大学生容易产生负面情绪，影响心理健康。通过在饮食中选择具有情绪调节作用的食物，可以实现改善大学生情绪，促进心理健康的目。

亲爱的同学们，健康的饮食习惯是我们幸福生活的基础。通过遵循膳食平衡基本原则，搭配丰富的食物种类，适量摄入营养素，有助于提高生活质量。让我们从现在开始，关注自己的饮食结构，积极践行健康饮食和平衡膳食的理念，为身体健康和学业进步打下坚实的基础。

愿大家都能拥有健康的体魄，充沛的精力，迎接每一天的挑战！

祝大家健康快乐！

某高校学生会

六、模型的评价与推广

6.1 模型的优点

1. 利用 EXCEL 软件对数据进行处理，作出各种图表，使结论简便、直观、快捷；
2. 运用了多种数学工具，互为参考，使计算结果更加准确，具有更高的可信度
3. 使用了线性规划模型对数据的处理速度较快，在数据量大的情况下尽可能节约时间成本，依据系数对每个变量都可以给出合理的解释，使数据更加形象。
4. 利用多目标规划模型更全面地考虑问题，同时考虑了多个目标，能够更好地反映实际问题的复杂性。提供多个备选解，可以在比较后进行选择，使计算结果更加有说服力，更好地满足数据需求。

6.2 模型的缺点

1. 线性规划模型对非线性数据的拟合效果不佳，对数据要进行初步判断是否满足线性关系。

2. 多目标规划模型在对数据的处理中计算难度较大，时间成本高。不同目标值之间可能出现冲突，较难寻找最优解。

6.3 模型的推广

基于该优化大学生饮食的模型，我们可以将该模型推广应用到中国公民的饮食建议，基于每个公民的具体情况给出最佳的饮食建议，给予每一位中国公民健康的体魄。

参考文献

- [1] 席萌菲, 孙茜宇. 大学生饮食营养与健康提升策略 [J]. 食品安全导刊, 2023(31):111-113. DOI:10.16043/j.cnki.cfs.2023.31.029
- [2] 柏胜. 高校餐厅对大学生日常饮食习惯的影响研究 [J]. 江苏调味副食品, 2023(04):38-41. DOI:10.16782/j.cnki.32-1235/ts.2023.04.010..
- [3] 张泽波, 王喆, 林梓宇, 等. 大学生的膳食与营养状况研究 [J]. 食品安全导刊, 2019(03):158+168. DOI:10.16043/j.cnki.cfs.2019.03.137.
- [4] 张亚迪, 柳莹, 王晨熙. 平衡膳食维持健康 [J]. 食品安全导刊, 2019(32):29. DOI:10.16043/j.cnki.cfs.2019.32.014.
- [5] 《中国食物成分表》(标准版, 第6版/第一册) (北京大学医学出版社)
- [6] 《中国食物成分表》(标准版, 第6版/第二册) (北京大学医学出版社)

附录 1 男大学生及女大学生的周食谱例

男大学生兼顾蛋白质氨基酸评分及经济性一周食谱			女大学生兼顾蛋白质氨基酸评分及经济性一周食谱		
1	2	3	1	2	3
k=1			k=1		
牛奶	大米饭	鸡排面	牛奶	大米饭	鸡排面
小米粥	红烧牛肉面	水饺	小米粥	红烧牛肉面	水饺
煮鸡蛋	拌菠菜	拌干豆腐	煮鸡蛋	拌菠菜	拌干豆腐
苹果	橙子	葡萄	橙子	苹果	葡萄
k=2			k=2		
酸奶	包子	鸡蛋饼	酸奶	包子	鸡蛋饼
南瓜粥	砂锅面	馄饨	南瓜粥	砂锅面	馄饨
煎鸡蛋	拌海带丝	拌木耳	煮鸡蛋	拌海带丝	拌木耳
香蕉	西瓜	葡萄	香蕉	西瓜	葡萄
k=3			k=3		
豆浆	大米饭	鸡排面	豆浆	大米饭	鸡排面
大米粥	红烧肉	馄饨面	大米粥	红烧肉	馄饨面
煮鸡蛋	拌芹菜花生米	拌土豆丝	煎鸡蛋	拌芹菜花生米	拌土豆丝
橙子	苹果	西瓜	橙子	苹果	西瓜
k=4			k=4		
牛奶	包子	鸡蛋饼	牛奶	包子	水饺
小米粥	砂锅面	水饺	小米粥	砂锅面	拌干豆腐
煎鸡蛋	拌干豆腐	拌菠菜	煮鸡蛋	拌菠菜	葡萄
香蕉	葡萄	柚子	香蕉	橙子	柚子
k=5			k=5		
酸奶	大米饭	鸡排面	酸奶	大米饭	鸡排面
大米粥	红烧牛肉面	馄饨	大米粥	红烧牛肉面	馄饨
煮鸡蛋	拌木耳	拌海带丝	煮鸡蛋	拌木耳	拌海带丝
苹果	橙子	西瓜	苹果	橙子	西瓜
k=6			k=6		
豆浆	包子	水饺	豆浆	包子	鸡蛋饼
南瓜粥	红烧牛肉面	拌干豆腐	南瓜粥	砂锅面	馄饨面
煎鸡蛋	拌芹菜花生米	葡萄	煎鸡蛋	拌芹菜花生米	拌干豆腐
橙子	西瓜	柚子	香蕉	苹果	葡萄
k=7			k=7		
牛奶	大米饭	鸡排面	牛奶	大米饭	水饺
小米粥	红烧肉	馄饨面	小米粥	红烧肉	拌菠菜
煮鸡蛋	拌土豆丝	拌菠菜	煮鸡蛋	拌土豆丝	柚子
苹果	橙子	葡萄	橙子	西瓜	-

附录 2 python 程序

男生蛋白质氨基酸评分的评分 python 程序

```
import pandas as pd  
import numpy as np
```

```
# 读取 ALL.xlsx 文件，获取食物的氨基酸含量数据  
all_data = pd.read_excel('ALL.xlsx')
```

```
# 检查 ALL 数据格式
```

```
print("ALL.xlsx 内容:")  
print(all_data.head())  
print(all_data.dtypes)
```

```
# 定义每餐食谱（使用主要成分）
```

```
breakfast = [  
    ('小米粥', '小米', 15, 1),  
    ('油条', '小麦粉', 50, 2),  
    ('油条', '豆油', 10, 2),  
    ('煎鸡蛋', '鸡蛋', 50, 1),  
    ('煎鸡蛋', '豆油', 10, 1),  
    ('拌海带丝', '海带', 100, 1),  
    ('拌海带丝', '芝麻油', 2, 1)  
]
```

```
lunch = [  
    ('大米饭', '稻米', 25, 4),  
    ('拌木耳', '木耳', 100, 1),  
    ('拌木耳', '芝麻油', 2, 1),  
    ('地三鲜', '茄子', 80, 1),  
    ('地三鲜', '土豆', 80, 1),  
    ('地三鲜', '青椒', 10, 1),  
    ('地三鲜', '豆油', 10, 1),  
    ('红烧肉', '五花猪肉', 50, 1),  
    ('红烧肉', '干豆腐', 50, 1),  
    ('红烧肉', '豆油', 10, 1)  
]
```

```
dinner = [  
    ('砂锅面', '玉米面', 80, 1),  
    ('砂锅面', '白菜', 20, 1),  
    ('砂锅面', '油菜', 20, 1),  
    ('砂锅面', '干豆腐', 10, 1),  
    ('砂锅面', '豆油', 10, 1),
```

```

('包子', '小麦粉', 25, 1),
('包子', '猪肉', 15, 1),
('包子', '酸菜', 20, 1),
('包子', '豆油', 5, 1),
('炸鸡块', '炸鸡块', 100, 1)
]

def get_food_aa_content(ingredient, portion, servings):
    food_row = all_data[all_data['主要成分'] == ingredient]
    if food_row.empty:
        print(f"警告: 主要成分 '{ingredient}' 在 ALL.xlsx 中未找到")
        return np.zeros(len(all_data.columns) - 1, dtype=float)
    aa_content = food_row.iloc[0, 1:] * portion * servings / 100 # 假设每列是每 100 克含量
    aa_content = aa_content.astype(float) # 确保氨基酸含量为浮点型
    print(f"主要成分: {ingredient}, 氨基酸含量: {aa_content.values}")
    return aa_content.values # 返回 numpy 数组

# 计算每餐的氨基酸含量总和
def calculate_meal_aa(meal):
    meal_aa = np.zeros(len(all_data.columns) - 1, dtype=float)
    for food in meal:
        food_aa = get_food_aa_content(food[1], food[2], food[3])
        if food_aa is not None:
            meal_aa += food_aa
    return meal_aa

breakfast_aa = calculate_meal_aa(breakfast)
lunch_aa = calculate_meal_aa(lunch)
dinner_aa = calculate_meal_aa(dinner)

# 打印氨基酸含量结果以进行调试
print("早餐氨基酸含量: ", breakfast_aa)
print("午餐氨基酸含量: ", lunch_aa)
print("晚餐氨基酸含量: ", dinner_aa)

# 定义参考蛋白质氨基酸评分模式
reference_aa = {
    '异亮氨酸': 40,
    '亮氨酸': 70,
    '赖氨酸': 55,
    '含硫氨基酸': 35,
    '芳香族氨基酸': 60,
    '苏氨酸': 40,
}

```

```

        '色氨酸': 10,
        '缬氨酸': 50
    }

# 计算氨基酸评分 (AAS)
def calculate_aas(aa_content):
    aas_scores = {}
    for aa in reference_aa.keys():
        if aa in all_data.columns:
            aa_index = all_data.columns.get_loc(aa) - 1
            aas_scores[aa] = aa_content[aa_index] / reference_aa[aa]
        else:
            print(f"警告: 氨基酸 '{aa}' 在 ALL.xlsx 中未找到")
            aas_scores[aa] = 0
    # 返回最小值即为第一限制氨基酸的评分值
    return min(aas_scores.values())

breakfast_aas = calculate_aas(breakfast_aa)
lunch_aas = calculate_aas(lunch_aa)
dinner_aas = calculate_aas(dinner_aa)

print(f'早餐氨基酸评分: {breakfast_aas}')
print(f'午餐氨基酸评分: {lunch_aas}')
print(f'晚餐氨基酸评分: {dinner_aas}')

```

女生蛋白质氨基酸评分的评分 python 程序

```

import pandas as pd
import numpy as np

# 读取 ALL.xlsx 文件, 获取食物的氨基酸含量数据
all_data = pd.read_excel('ALL.xlsx')

# 定义每餐食谱 (使用主要成分)
breakfast = [
    ('豆浆', '黄豆', 10, 1),
    ('鸡排面', '小麦粉', 50, 1),
    ('鸡排面', '鸡肉', 40, 1),
    ('鸡排面', '豆油', 5, 1)
]

lunch = [
    ('鸡蛋饼', '小麦粉', 25, 1),
    ('鸡蛋饼', '鸡蛋', 20, 1),
    ('鸡蛋饼', '火腿肠', 20, 1),
]

```

```

        ('鸡蛋饼', '豆油', 5, 1),
        ('水饺', '小麦粉', 50, 1),
        ('水饺', '猪肉', 20, 1),
        ('水饺', '白菜', 40, 1),
        ('水饺', '豆油', 10, 1),
        ('葡萄', '葡萄', 100, 1)
    ]
}

dinner = [
    ('大米饭', '稻米', 25, 2),
    ('香菇炒油菜', '油菜', 100 * 0.5, 1),
    ('香菇炒油菜', '香菇', 20, 1),
    ('香菇炒油菜', '豆油', 5, 1),
    ('炒肉蒜台', '蒜台', 100 * 0.5, 1),
    ('炒肉蒜台', '猪肉', 30, 1),
    ('炒肉蒜台', '豆油', 5, 1),
    ('茄汁沙丁鱼', '茄汁沙丁鱼', 100 * 0.5, 1),
    ('苹果', '苹果', 100, 1)
]
]

def get_food_aa_content(ingredient, portion, servings):
    food_row = all_data[all_data['主要成分'] == ingredient]
    if food_row.empty:
        print(f"警告: 主要成分 '{ingredient}' 在 ALL.xlsx 中未找到")
        return np.zeros(len(all_data.columns) - 1, dtype=float)
    aa_content = food_row.iloc[0, 1:] * portion * servings / 100 # 假设每列是每 100 克含量
    aa_content = aa_content.astype(float) # 确保氨基酸含量为浮点型
    print(f"主要成分: {ingredient}, 氨基酸含量: {aa_content.values}")
    return aa_content.values # 返回 numpy 数组

# 计算每餐的氨基酸含量总和
def calculate_meal_aa(meal):
    meal_aa = np.zeros(len(all_data.columns) - 1, dtype=float)
    for food in meal:
        food_aa = get_food_aa_content(food[1], food[2], food[3])
        if food_aa is not None:
            meal_aa += food_aa
    return meal_aa

breakfast_aa = calculate_meal_aa(breakfast)
lunch_aa = calculate_meal_aa(lunch)
dinner_aa = calculate_meal_aa(dinner)

```

```

# 打印氨基酸含量结果以进行调试
print("早餐氨基酸含量: ", breakfast_aa)
print("午餐氨基酸含量: ", lunch_aa)
print("晚餐氨基酸含量: ", dinner_aa)

# 定义参考蛋白质氨基酸评分模式
reference_aa = {
    '异亮氨酸': 40,
    '亮氨酸': 70,
    '赖氨酸': 55,
    '含硫氨基酸': 35,
    '芳香族氨基酸': 60,
    '苏氨酸': 40,
    '色氨酸': 10,
    '缬氨酸': 50
}

# 计算氨基酸评分 (AAS)
def calculate_aas(aa_content):
    aas_scores = {}
    for aa in reference_aa.keys():
        if aa in all_data.columns:
            aa_index = all_data.columns.get_loc(aa) - 1
            aas_scores[aa] = aa_content[aa_index] / reference_aa[aa]
        else:
            print(f"警告: 氨基酸 '{aa}' 在 ALL.xlsx 中未找到")
            aas_scores[aa] = 0
    # 返回最小值即为第一限制氨基酸的评分值
    return min(aas_scores.values())

breakfast_aas = calculate_aas(breakfast_aa)
lunch_aas = calculate_aas(lunch_aa)
dinner_aas = calculate_aas(dinner_aa)

print(f'早餐氨基酸评分: {breakfast_aas}')
print(f'午餐氨基酸评分: {lunch_aas}')
print(f'晚餐氨基酸评分: {dinner_aas}')

```

男女食谱优化后的蛋白质氨基酸评分

```

import pandas as pd
import numpy as np
from scipy.optimize import linprog

# 读取数据

```

```

all_data = pd.read_excel('ALL.xlsx')
meals_data = pd.read_excel('食堂数据.xlsx')

# 假设费用信息在 meals_data 中有一列 "费用"
# 如果没有, 请根据实际情况添加费用信息
# 创建食物和费用的映射表
food_costs = {
    '小米粥': 2, '油条': 1.5, '煎鸡蛋': 1, '拌海带丝': 2, '大米饭': 3,
    '拌木耳': 2, '地三鲜': 3, '红烧肉': 5, '砂锅面': 4, '包子': 1,
    '炸鸡块': 4, '豆浆': 2, '鸡排面': 4, '鸡蛋饼': 3, '水饺': 5,
    '葡萄': 3, '香菇炒油菜': 2.5, '炒肉蒜台': 3, '茄汁沙丁鱼': 5,
    '苹果': 2
}

# 获取每种食物的主要成分和氨基酸含量
def get_food_aa_content(ingredient, portion, servings):
    food_row = all_data[all_data['主要成分'] == ingredient]
    if food_row.empty:
        return np.zeros(len(all_data.columns) - 1, dtype=float)
    aa_content = food_row.iloc[0, 1:] * portion * servings / 100
    aa_content = aa_content.astype(float)
    return aa_content.values

# 定义食谱
breakfast = [('豆浆', '黄豆', 10, 1), ('鸡排面', '小麦粉', 50, 1), ('鸡排面', '鸡肉', 40, 1), ('鸡排面', '豆油', 5, 1)]
lunch = [('鸡蛋饼', '小麦粉', 25, 1), ('鸡蛋饼', '鸡蛋', 20, 1), ('鸡蛋饼', '火腿肠', 20, 1), ('鸡蛋饼', '豆油', 5, 1),
          ('水饺', '小麦粉', 50, 1), ('水饺', '猪肉', 20, 1), ('水饺', '白菜', 40, 1), ('水饺', '豆油', 10, 1),
          ('葡萄', '葡萄', 100, 1)]
dinner = [('大米饭', '稻米', 25, 2), ('香菇炒油菜', '油菜', 50, 1), ('香菇炒油菜', '香菇', 20, 1), ('香菇炒油菜', '豆油', 5, 1),
          ('炒肉蒜台', '蒜台', 50, 1), ('炒肉蒜台', '猪肉', 30, 1), ('炒肉蒜台', '豆油', 5, 1), ('茄汁沙丁鱼', '茄汁沙丁鱼', 50, 1),
          ('苹果', '苹果', 100, 1)]

all_meals = breakfast + lunch + dinner

# 获取食谱中的所有主要成分
ingredients = [food[1] for food in all_meals]

# 获取氨基酸含量和费用
aa_contents = np.array([get_food_aa_content(food[1], food[2], food[3]) for

```

```

food in all_meals])
costs = np.array([food_costs[food[0]] for food in all_meals])

# 目标 1: 最大化氨基酸评分
c = -aa_contents.sum(axis=1)
result1 = linprog(c, bounds=[(0, 10)] * len(all_meals)) # 假设每种食物
最多吃 10 份

# 目标 2: 最小化费用
c = costs
result2 = linprog(c, bounds=[(0, 10)] * len(all_meals))

# 目标 3: 兼顾氨基酸评分和费用
lambda_ = 0.5
c = -lambda_ * aa_contents.sum(axis=1) + (1 - lambda_) * costs
result3 = linprog(c, bounds=[(0, 10)] * len(all_meals))

# 计算膳食营养评分
def calculate_nutrition_score(result, aa_contents):
    x = result.x
    total_aa_content = (aa_contents.T * x).sum(axis=1)
    # 计算氨基酸评分（假设氨基酸在第一个 8 列）
    aa_scores = total_aa_content[:8] / np.array([40, 70, 55, 35, 60, 40,
10, 50])
    return aa_scores.min() * 100 # 取最小值作为评分

# 评价
score1 = calculate_nutrition_score(result1, aa_contents)
score2 = calculate_nutrition_score(result2, aa_contents)
score3 = calculate_nutrition_score(result3, aa_contents)

print(f"目标 1 - 氨基酸评分最大化: 评分 = {score1}")
print(f"目标 2 - 费用最经济: 评分 = {score2}")
print(f"目标 3 - 兼顾氨基酸评分和费用: 评分 = {score3}")

```

求解第二问的 python 程序

```

import pandas as pd
import numpy as np
from scipy.optimize import linprog

# 读取数据
all_data = pd.read_excel('ALL.xlsx')
all_data.set_index('主要成分', inplace=True)

```

```

food_nutrition = pd.read_excel('食堂数据.xlsx')
food_nutrition.set_index('食物名称', inplace=True)

# 检查数据并填充缺失值
all_data = all_data.fillna(0)
food_nutrition = food_nutrition.fillna(0)

# 参考蛋白质氨基酸评分模式
reference_aa = {
    '异亮氨酸': 40,
    '亮氨酸': 70,
    '赖氨酸': 55,
    '含硫氨基酸': 35,
    '芳香族氨基酸': 60,
    '苏氨酸': 40,
    '色氨酸': 10,
    '缬氨酸': 50
}

def calculate_aas(aa_content, servings):
    total_aa_intake = aa_content.mul(servings, axis=0).sum(axis=0)
    reference_intake = np.array(list(reference_aa.values()))
    aa_scores = total_aa_intake / reference_intake
    score = (np.sum(total_aa_intake) / np.sum(reference_intake)) * 100
    return score

def optimize_protein(gender):
    aa_columns = list(reference_aa.keys())
    aa_content = all_data[aa_columns].reindex(food_nutrition.index).fillna(0)
    c = -np.ones(len(food_nutrition))

    if gender == '男':
        daily_needs = {
            '蛋白质/g': (55, 75), # 设置范围
            '能量/kcal': (2200, 2600),
            '脂肪/g': (60, 80),
            '碳水化合物/g': (270, 330)
        }
    else:
        daily_needs = {
            '蛋白质/g': (45, 65), # 设置范围
            '能量/kcal': (1700, 2100),
        }

    return daily_needs

```

```

        '脂肪/g': (50, 70),
        '碳水化合物/g': (220, 280)
    }

A_ub = [
    food_nutrition['蛋白质/g'].values,
    food_nutrition['能量/kcal'].values,
    food_nutrition['脂肪/g'].values,
    food_nutrition['碳水化合物/g'].values
]
b_ub = [upper for _, upper in daily_needs.values()]

A_eq = [
    food_nutrition['蛋白质/g'].values,
    food_nutrition['能量/kcal'].values,
    food_nutrition['脂肪/g'].values,
    food_nutrition['碳水化合物/g'].values
]
b_eq = [lower for lower, _ in daily_needs.values()]

x_bounds = [(0, None) for _ in range(len(food_nutrition))]

print("目标函数:", c)
print("约束矩阵(不等式):", A_ub)
print("约束向量(不等式):", b_ub)
print("约束矩阵(等式):", A_eq)
print("约束向量(等式):", b_eq)
print("变量边界:", x_bounds)

result = linprog(c, A_ub=A_ub, b_ub=b_ub, A_eq=A_eq, b_eq=b_eq,
bounds=x_bounds, method='highs')

if result.success:
    servings = result.x
    optimized_diet = pd.DataFrame({
        '主要成分': food_nutrition.index,
        '摄入量/克': servings,
        '蛋白质/g': food_nutrition['蛋白质/g'] * servings,
        '能量/kcal': food_nutrition['能量/kcal'] * servings,
        '脂肪/g': food_nutrition['脂肪/g'] * servings,
        '碳水化合物/g': food_nutrition['碳水化合物/g'] * servings,
        '费用/元': food_nutrition['费用/元'] * servings
    })
    aa_score = calculate_aas(aa_content, servings)

```

```

        return optimized_diet, aa_score
    else:
        print("优化失败: ", result.message)
        return None, 0

# 进行优化
diet_protein_male, score_male = optimize_protein('男')
diet_protein_female, score_female = optimize_protein('女')

print("男生蛋白质优化日食谱:")
print(diet_protein_male)
print("男生蛋白质优化评分:", score_male)
print("女生蛋白质优化日食谱:")
print(diet_protein_female)
print("女生蛋白质优化评分:", score_female)

def optimize_cost(gender):
    c = food_nutrition['费用/元'].values

    if gender == '男':
        daily_needs = {
            '蛋白质/g': (55, 75), # 设置范围
            '能量/kcal': (2200, 2600),
            '脂肪/g': (60, 80),
            '碳水化合物/g': (270, 330)
        }
    else:
        daily_needs = {
            '蛋白质/g': (45, 65), # 设置范围
            '能量/kcal': (1700, 2100),
            '脂肪/g': (50, 70),
            '碳水化合物/g': (220, 280)
        }

    A_ub = [
        food_nutrition['蛋白质/g'].values,
        food_nutrition['能量/kcal'].values,
        food_nutrition['脂肪/g'].values,
        food_nutrition['碳水化合物/g'].values
    ]
    b_ub = [upper for _, upper in daily_needs.values()]

    A_eq = [
        food_nutrition['蛋白质/g'].values,

```

```

        food_nutrition['能量/kcal'].values,
        food_nutrition['脂肪/g'].values,
        food_nutrition['碳水化合物/g'].values
    ]
    b_eq = [lower for lower, _ in daily_needs.values()]

    x_bounds = [(0, None) for _ in range(len(food_nutrition))]

    print("目标函数:", c)
    print("约束矩阵(不等式):", A_ub)
    print("约束向量(不等式):", b_ub)
    print("约束矩阵(等式):", A_eq)
    print("约束向量(等式):", b_eq)
    print("变量边界:", x_bounds)

    result = linprog(c, A_ub=A_ub, b_ub=b_ub, A_eq=A_eq, b_eq=b_eq,
                      bounds=x_bounds, method='highs')

    if result.success:
        servings = result.x
        optimized_diet = pd.DataFrame({
            '主要成分': food_nutrition.index,
            '摄入量/克': servings,
            '蛋白质/g': food_nutrition['蛋白质/g'] * servings,
            '能量/kcal': food_nutrition['能量/kcal'] * servings,
            '脂肪/g': food_nutrition['脂肪/g'] * servings,
            '碳水化合物/g': food_nutrition['碳水化合物/g'] * servings,
            '费用/元': food_nutrition['费用/元'] * servings
        })
        return optimized_diet
    else:
        print("优化失败:", result.message)
        return None

# 进行优化
diet_cost_male = optimize_cost('男')
diet_cost_female = optimize_cost('女')

print("男生费用优化日食谱:")
print(diet_cost_male)
print("女生费用优化日食谱:")
print(diet_cost_female)

def evaluate_diet(diet):

```

```

if diet is None:
    return {
        '总蛋白质/g': 0,
        '总能量/kcal': 0,
        '总脂肪/g': 0,
        '总碳水化合物/g': 0,
        '总费用/元': 0,
        '氨基酸评分': 0
    }

total_protein = diet['蛋白质/g'].sum()
total_calories = diet['能量/kcal'].sum()
total_fat = diet['脂肪/g'].sum()
total_carbs = diet['碳水化合物/g'].sum()
total_cost = diet['费用/元'].sum()
aa_columns = list(reference_aa.keys())
aa_content = all_data[aa_columns].reindex(diet.index).fillna(0)
aa_score = calculate_aas(aa_content, diet['摄入量/克'])

return {
    '总蛋白质/g': total_protein,
    '总能量/kcal': total_calories,
    '总脂肪/g': total_fat,
    '总碳水化合物/g': total_carbs,
    '总费用/元': total_cost,
    '氨基酸评分': aa_score
}

# 评价食谱
evaluation_protein_male = evaluate_diet(diet_protein_male)
evaluation_protein_female = evaluate_diet(diet_protein_female)
evaluation_cost_male = evaluate_diet(diet_cost_male)
evaluation_cost_female = evaluate_diet(diet_cost_female)

print("男生费用优化营养评价:", evaluation_cost_male)
print("女生费用优化营养评价:", evaluation_cost_female)

comparison = pd.DataFrame({
    '指标': ['总蛋白质/g', '总能量/kcal', '总脂肪/g', '总碳水化合物/g', '总费用/元', '氨基酸评分'],
    '男生 蛋白质 优化': [evaluation_protein_male[key] for key in evaluation_protein_male],
    '男生 费用 优化': [evaluation_cost_male[key] for key in evaluation_cost_male],
})

```

```

    '女生蛋白质优化': [evaluation_protein_female[key] for key in
evaluation_protein_female],
    '女生费用优化': [evaluation_cost_female[key] for key in
evaluation_cost_female]
})

print("比较分析:")
print(comparison)

```

求解第三问的 python 程序

```

import pandas as pd
import numpy as np
from scipy.optimize import linprog
from scipy.optimize import Bounds

# 读取 ALL.xlsx 文件，获取食物的氨基酸含量数据
all_data = pd.read_excel('ALL.xlsx')
all_data.set_index('主要成分', inplace=True)

# 读取附件 3 中的食物营养数据
food_nutrition = pd.read_excel('食堂数据.xlsx')
food_nutrition.set_index('主要成分', inplace=True)

# 定义参考蛋白质氨基酸评分模式
reference_aa = {
    '异亮氨酸': 40,
    '亮氨酸': 70,
    '赖氨酸': 55,
    '含硫氨基酸': 35,
    '芳香族氨基酸': 60,
    '苏氨酸': 40,
    '色氨酸': 10,
    '缬氨酸': 50
}

def calculate_aas(aa_content, servings):
    """
    根据氨基酸评分计算公式计算蛋白质氨基酸评分。
    """

    total_aa_intake = aa_content.mul(servings, axis=0).sum(axis=0)
    reference_intake = np.array(list(reference_aa.values()))
    score = (np.sum(total_aa_intake) / np.sum(reference_intake)) * 100
    return score

```

```

def optimize_protein(gender):
    """
    目标函数：最大化蛋白质氨基酸评分
    """
    aa_columns = list(reference_aa.keys())
    aa_content = all_data[aa_columns].reindex(food_nutrition.index).fillna(0)

    # 目标函数：最大化氨基酸评分
    c = -np.ones(len(food_nutrition))

    # 约束条件：满足基本营养需求
    if gender == '男':
        daily_needs = {
            '蛋白质/g': 65,
            '能量/kcal': 2500,
            '脂肪/g': 70,
            '碳水化合物/g': 300
        }
    else:
        daily_needs = {
            '蛋白质/g': 55,
            '能量/kcal': 2000,
            '脂肪/g': 60,
            '碳水化合物/g': 250
        }

    A = [
        food_nutrition['蛋白质/g'].values,      # 蛋白质
        food_nutrition['能量/kcal'].values,      # 能量
        food_nutrition['脂肪/g'].values,         # 脂肪
        food_nutrition['碳水化合物/g'].values   # 碳水化合物
    ]
    b = [daily_needs['蛋白质/g'], daily_needs['能量/kcal'], daily_needs['脂肪/g'], daily_needs['碳水化合物/g']]

    # 每种食物的摄入量应大于等于 0
    x_bounds = [(0, None) for _ in range(len(food_nutrition))]

    # 求解线性规划问题
    result = linprog(c, A_ub=A, b_ub=b, bounds=x_bounds, method='highs')

    if result.success:
        servings = result.x

```

```

optimized_diet = pd.DataFrame({
    '主要成分': food_nutrition.index,
    '摄入量/克': servings,
    '蛋白质/g': food_nutrition['蛋白质/g'] * servings,
    '能量/kcal': food_nutrition['能量/kcal'] * servings,
    '脂肪/g': food_nutrition['脂肪/g'] * servings,
    '碳水化合物/g': food_nutrition['碳水化合物/g'] * servings,
    '费用/元': food_nutrition['费用/元'] * servings
})
aa_score = calculate_aas(aa_content, servings)
return optimized_diet, aa_score
else:
    print("优化失败")
    return None, 0

# 优化一周的食谱
def weekly_optimize_protein(gender):
    weekly_diet = []
    weekly_score = 0
    all_food_items = set()
    for day in range(7):
        daily_diet, daily_score = optimize_protein(gender)
        weekly_diet.append(daily_diet)
        weekly_score += daily_score
        all_food_items.update(daily_diet['主要成分'][daily_diet['摄入量/克'] > 0].values)

    # 确保摄入的食材种类超过 25 种
    if len(all_food_items) < 25:
        raise ValueError(f"食材种类不足 25 种, 仅有{len(all_food_items)}种")

    weekly_diet_df = pd.concat(weekly_diet, keys=['周一', '周二', '周三', '周四', '周五', '周六', '周日'])
    average_score = weekly_score / 7
    return weekly_diet_df, average_score

# 进行优化
weekly_diet_protein_male, weekly_score_male = weekly_optimize_protein('男')
weekly_diet_protein_female, weekly_score_female = weekly_optimize_protein('女')

print("男生蛋白质优化周食谱:")

```

```

print(weekly_diet_protein_male)
print("男生蛋白质优化评分:", weekly_score_male)
print("女生蛋白质优化周食谱:")
print(weekly_diet_protein_female)
print("女生蛋白质优化评分:", weekly_score_female)

#餐费
def optimize_cost(gender):
    """
    目标函数: 最小化用餐费用
    """
    # 目标函数: 最小化用餐费用
    c = food_nutrition['费用/元'].values

    # 约束条件: 满足基本营养需求
    if gender == '男':
        daily_needs = {
            '蛋白质/g': 65,
            '能量/kcal': 2500,
            '脂肪/g': 70,
            '碳水化合物/g': 300
        }
    else:
        daily_needs = {
            '蛋白质/g': 55,
            '能量/kcal': 2000,
            '脂肪/g': 60,
            '碳水化合物/g': 250
        }

    A = [
        food_nutrition['蛋白质/g'].values,      # 蛋白质
        food_nutrition['能量/kcal'].values,      # 能量
        food_nutrition['脂肪/g'].values,         # 脂肪
        food_nutrition['碳水化合物/g'].values   # 碳水化合物
    ]
    b = [daily_needs['蛋白质/g'], daily_needs['能量/kcal'], daily_needs['脂肪/g'],
          daily_needs['碳水化合物/g']]

    # 每种食物的摄入量应大于等于 0
    x_bounds = [(0, None) for _ in range(len(food_nutrition))]
```

```

# 求解线性规划问题
result = linprog(c, A_ub=A, b_ub=b, bounds=x_bounds, method='highs')

if result.success:
    servings = result.x
    optimized_diet = pd.DataFrame({
        '主要成分': food_nutrition.index,
        '摄入量/克': servings,
        '蛋白质/g': food_nutrition['蛋白质/g'] * servings,
        '能量/kcal': food_nutrition['能量/kcal'] * servings,
        '脂肪/g': food_nutrition['脂肪/g'] * servings,
        '碳水化合物/g': food_nutrition['碳水化合物/g'] * servings,
        '费用/元': food_nutrition['费用/元'] * servings
    })
    return optimized_diet
else:
    print("优化失败")
    return None

# 优化一周的食谱
def weekly_optimize_cost(gender):
    weekly_diet = []
    all_food_items = set()
    for day in range(7):
        daily_diet = optimize_cost(gender)
        weekly_diet.append(daily_diet)
        all_food_items.update(daily_diet['主要成分'][daily_diet['摄入量/克'] > 0].values)

    # 确保摄入的食材种类超过 25 种
    if len(all_food_items) < 25:
        raise ValueError(f"食材种类不足 25 种，仅有{len(all_food_items)}种")

    weekly_diet_df = pd.concat(weekly_diet, keys=['周一', '周二', '周三', '周四', '周五', '周六', '周日'])
    total_cost = weekly_diet_df['费用/元'].sum()
    return weekly_diet_df, total_cost

# 进行优化
weekly_diet_cost_male, total_cost_male = weekly_optimize_cost('男')
weekly_diet_cost_female, total_cost_female = weekly_optimize_cost('女')

print("男生费用优化周食谱:")

```

```

print(weekly_diet_cost_male)
print("男生费用总计:", total_cost_male)
print("女生费用优化周食谱:")
print(weekly_diet_cost_female)
print("女生费用总计:", total_cost_female)

#兼顾
def optimize_combined(gender, alpha=0.5):
    """
    目标函数: 兼顾蛋白质氨基酸评分及用餐费用
    """
    aa_columns = list(reference_aa.keys())
    aa_content = all_data[aa_columns].reindex(food_nutrition.index).fillna(0)

    # 目标函数: 最小化费用, 同时最大化氨基酸评分
    cost = food_nutrition['费用/元'].values
    aa_score = np.ones(len(food_nutrition))

    # 将目标函数转化为一个加权的目标函数
    c = alpha * cost - (1 - alpha) * aa_score

    # 约束条件: 满足基本营养需求
    if gender == '男':
        daily_needs = {
            '蛋白质/g': 65,
            '能量/kcal': 2500,
            '脂肪/g': 70,
            '碳水化合物/g': 300
        }
    else:
        daily_needs = {
            '蛋白质/g': 55,
            '能量/kcal': 2000,
            '脂肪/g': 60,
            '碳水化合物/g': 250
        }

    A = [
        food_nutrition['蛋白质/g'].values,      # 蛋白质
        food_nutrition['能量/kcal'].values,      # 能量
        food_nutrition['脂肪/g'].values,         # 脂肪
        food_nutrition['碳水化合物/g'].values   # 碳水化合物
    ]

```

```

        ]
        b = [daily_needs['蛋白质/g'], daily_needs['能量/kcal'], daily_needs['
脂肪/g'], daily_needs['碳水化合物/g']]

# 每种食物的摄入量应大于等于 0
x_bounds = [(0, None) for _ in range(len(food_nutrition))]

# 求解线性规划问题
result = linprog(c, A_ub=A, b_ub=b, bounds=x_bounds, method='highs')

if result.success:
    servings = result.x
    optimized_diet = pd.DataFrame({
        '主要成分': food_nutrition.index,
        '摄入量/克': servings,
        '蛋白质/g': food_nutrition['蛋白质/g'] * servings,
        '能量/kcal': food_nutrition['能量/kcal'] * servings,
        '脂肪/g': food_nutrition['脂肪/g'] * servings,
        '碳水化合物/g': food_nutrition['碳水化合物/g'] * servings,
        '费用/元': food_nutrition['费用/元'] * servings
    })
    aa_score = calculate_aas(aa_content, servings)
    return optimized_diet, aa_score
else:
    print("优化失败")
    return None, 0

# 优化一周的食谱
def weekly_optimize_combined(gender, alpha=0.5):
    weekly_diet = []
    weekly_score = 0
    all_food_items = set()
    for day in range(7):
        daily_diet, daily_score = optimize_combined(gender, alpha)
        weekly_diet.append(daily_diet)
        weekly_score += daily_score
        all_food_items.update(daily_diet['主要成分'][daily_diet['摄入量
/克'] > 0].values)

    # 确保摄入的食材种类超过 25 种
    if len(all_food_items) < 25:
        raise ValueError(f"食材种类不足 25 种，仅有{len(all_food_items)}种")

```

```

    weekly_diet_df = pd.concat(weekly_diet, keys=['周一', '周二', '周三',
'周四', '周五', '周六', '周日'])
    average_score = weekly_score / 7
    return weekly_diet_df, average_score

# 进行优化
    weekly_diet_combined_male, weekly_score_combined_male =
    weekly_optimize_combined('男')
    weekly_diet_combined_female, weekly_score_combined_female =
    weekly_optimize_combined('女')

    print("男生综合优化周食谱:")
    print(weekly_diet_combined_male)
    print("男生综合优化评分:", weekly_score_combined_male)
    print("女生综合优化周食谱:")
    print(weekly_diet_combined_female)
    print("女生综合优化评分:", weekly_score_combined_female)

#分析
comparison_weekly = pd.DataFrame({
    '指标': ['总蛋白质/g', '总能量/kcal', '总脂肪/g', '总碳水化合物/g', '总费用/元', '氨基酸评分'],
    '男生蛋白质优化': [weekly_diet_protein_male[key].sum() if key != '氨基酸评分' else weekly_score_male for key in weekly_diet_protein_male.columns],
    '男生费用优化': [weekly_diet_cost_male[key].sum() if key != '氨基酸评分' else np.nan for key in weekly_diet_cost_male.columns],
    '男生综合优化': [weekly_diet_combined_male[key].sum() if key != '氨基酸评分' else weekly_score_combined_male for key in weekly_diet_combined_male.columns],
    '女生蛋白质优化': [weekly_diet_protein_female[key].sum() if key != '氨基酸评分' else weekly_score_female for key in weekly_diet_protein_female.columns],
    '女生费用优化': [weekly_diet_cost_female[key].sum() if key != '氨基酸评分' else np.nan for key in weekly_diet_cost_female.columns],
    '女生综合优化': [weekly_diet_combined_female[key].sum() if key != '氨基酸评分' else weekly_score_combined_female for key in weekly_diet_combined_female.columns]
})

    print("比较分析:")
    print(comparison_weekly)

```