影响城市居民身体健康的因素分析

摘要

本文基于一份某市针对部分居民所做的“慢性非传染性疾病及其相关影响因素流行病学”调查结果和最新修订的《中国居民膳食指南》，探讨居民的饮食生活习惯及其与年龄、性别等个人因素之间的关系，进一步深入研究慢性疾病与饮食生活习惯等因素的相关性，寻找影响城市居民身体健康的因素，最终对居民进行分类，并针对不同人群提出有利于身体健康的建议。

由于样本数据规模大、维度高，为便于后续分析，首先我们基于**用户画像**的思想，利用 python 将每一个调查对象抽象为一个 **PERSON** 类，利用类的嵌套对所有信息进行分类存储与处理，筛选并处理矛盾、空缺等问题数据。

针对问题一，我们根据附件 A3 的八大准则及调查内容提炼出 **13 条指标**，逐一统计计算每条指标的满足率，并结合具体的数据分布直方图，对全体调查对象饮食习惯的合理性做出评估，指出存在的 4 个主要问题。

针对问题二，我们首先在问题一的基础上增加 2 条关于生活习惯的指标，形成饮食习惯指标组和生活习惯指标组，再分别与年龄、性别等个人基本信息组成的指标组进行**典型相关分析**，得到指标间的相关信息。

针对问题三，根据调查所得信息，确定五种慢性病分析对象为：高血糖、高血压、高血脂、肥胖、高尿酸。首先提取是否患上述五种疾病为定类指标（健康、易患病、患病），然后在前两问的基础上添加一条吸烟到影响因素指标组，将其与疾病指标组进行**典型相关分析**，初步得到影响每个慢性病的几个主要因素。接下来利用 **XGBOOST-SHAP 模型**得到五种慢性病与吸烟、饮酒、饮食习惯、生活习惯、工作性质、运动等因素的关系及相关程度。

针对问题四，选择两个指标作为分类依据。首先，依据问题三结果和专业医学知识，将五种慢性病合并为一个指标。其次，对患病指标和年龄指标进行**卡方检验**，得出年龄对患病情况有显著影响的结论，从而将年龄作为另一个划分指标。据此将所有居民分成九类。最后，分别对不同年龄段的群体重新利用 **XGBOOST-SHAP 模型**分析患病情况与生活饮食等习惯之间的关系，据此给出有利于不同群体身体健康的膳食、运动等方面的建议。

关键字：典型相关分析　XGBOOST-SHAP 模型　卡方检验　偏相关分析

一、 问题背景与重述

1.1 问题背景

随着人们现代生活方式的变化，慢性非传染性疾病（以下简称慢性病）的患病率持续增加，其中包括心脑血管疾病、糖尿病、恶性肿瘤和慢性阻塞性肺病等。这些疾病已经成为影响我国居民身体健康的重要因素。由于身体健康状况与年龄、饮食习惯、身体活动水平、职业等因素密切相关，因此，如何通过科学合理的膳食、适度的体育锻炼以及积极践行健康的生活方式，以促进身体健康，已成为社会各界普遍关心的重要议题。

1.2 问题提出

现有某市卫生健康研究部门对部分居民所做的“慢性非传染性疾病及其相关影响因素流行病学”调查的数据结果，以中国营养学会最新修订的《中国居民膳食指南》为准则，尝试解决以下问题：

(1) **问题一**：参考《中国居民膳食指南》的八条准则，分析所调查的居民饮食习惯的合理性，并说明存在的主要问题。
(2) **问题二**：分析居民的生活习惯和饮食习惯是否与年龄、性别、婚姻状况、文化程度、职业等因素相关。
(3) **问题三**：基于调查数据，对常见慢性疾病（例如高血压、糖尿病等）与吸烟、饮酒、饮食习惯、生活习惯、工作性质、运动等因素之间的关系及其相关程度进行深入分析。
(4) **问题四**：依据具体情况对被调查的居民进行分类，并针对各类人群提出有利于身体健康的膳食、运动等方面的合理建议。

二、 数据预处理

调查问卷共收集了 7836 份数据，数据规模大、维度高，为便于后续分析研究，首先将调查数据导入到 python 程序中。

2.1 数据处理的思想

由于每一行数据都可以表征一个人的特征，因此我们基于**用户画像**的思想，将每一个个体的调查数据抽象为一个 **PERSON** 类。

在设计 PERSON 类的时候，考虑到每个调查对象都有 234 个数据需要记录，我们采用了类的嵌套的形式进行数据的分类处理，以达到逻辑上清晰。下图与下表是我们对

PERSON 类的设计。

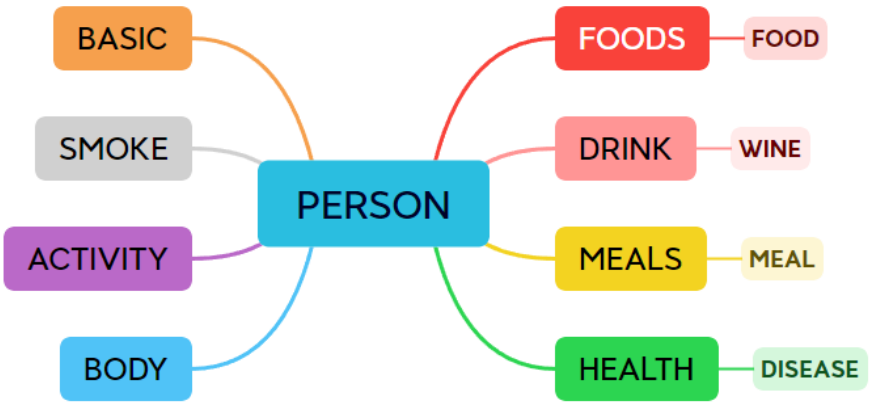


图 1　PERSON 类设计

表 1　PERSON 类详细介绍

名称	内容	属性	子类	子类内容	对应的列
BASIC	基本情况	ID、性别等	无	-	001-008
SMOKE	吸烟情况	是否吸烟等	无	-	009-014
DRINK	饮酒情况	是否饮酒等	WINE	频率用量等	015-031
MEALS	三餐情况	早中晚餐等	MEAL	地点人数等	032-052
FOODS	饮食情况	各种食物等	FOOD	频率用量等	053-194
ACTIVITY	身体活动	工作锻炼等	无	-	195-199
HEALTH	健康情况	各种疾病等	DISEASE	时间措施等	200-221
BODY	体检情况	身体数据等	无	-	222-234

2.2 无效数据的处理

在后续数据的处理过程中，我们发现了一部分人在填写问卷时出现如填写数据前后矛盾，数据漏填等问题，以下是我们发现的具有代表性的无效数据以及处理方式（这里列出四项，其余见附录）：

表 2　无效数据处理

所属类别	内容	问题 ID	处理方式
BASIC	部分人的 ID 遗失	17836 等	保持原始顺序下补充
BASIC	未说明民族情况	10232 等	在统计时标记为 None
...	...	...	...
ACTIVITY	锻炼时间未填写	10019 等	在统计时标记为 None
BODY	身高体重未填写	13715 等	在统计时标记为 None

三、 符号和变量说明

符号或变量	意义
$born$	出生年
age	年龄
$drink\_freq$	喝酒频率（次/周）
$drink\_volume$	每次喝酒的量（两）
$wine\_degree$	酒精度数
$alcohol\_gram$	酒精克数
eat_freq	食物食用频率（换算成次/天）
eat_con	平均每次食用量（两）
eat_per_day	平均每天食用量（克）
oil_per_month	烹饪油的全家食用量（斤/月）
num_people	平均每餐人数
oil_per_day	每天食用烹饪油量（克）
$height$	身高（厘米）
$weight$	体重（千克）
BMI	BMI 值

四、 模型假设

4.1 计算常量假设

1. 一个月在计算时假设为 30 天。
2. 依据第七次全国人口普查数据，认为平均每个家庭户的人口为 2.62 人，将以家庭为单位询问的数据项目由全家食用量折合成个人食用量。
3. 在计算摄入酒精量时，将高度白酒、低度白酒、啤酒、黄酒、葡萄酒的度数分别假设为 50、40、4、15、10 度。
4. 在计算豆制品的时候，50g 大豆 = 200g 豆腐 = 80g 豆腐丝 = 730g 豆浆

4.2 计算公式假设

1. BASIC：$age = 2023 - born$
2. DRINK：$alcohol\_gram = \frac{drink\_freq}{7} * (drink\_volume * 50) * 0.8 * \frac{wine\_degree}{100}$
3. FOODS：$eat\_per\_day = eat\_freq * eat\_con * 50$
4. FOODS：$oil\_per\_day = \frac{oil\_per\_month}{30} * 500/num\_people$
5. BODY：$BMI = weight/(\frac{height}{100})^2$

4.3 食物类别假设

1. 谷薯类：D4、D5、D6、D7、D8、D27
2. 蔬果类：D22、D23、D24、D25、D26、D28、D29
3. 畜禽鱼蛋奶：D9、D10、D11、D12、D13、D14、D15、D16、D17
4. 豆类：D18、D19、D20、D21
5. 奶制品：D14、D15、D16
6. 全谷物：D5、D6

4.4 疾病标准假设

表 3　高脂血症诊断标准假设（满足其一即患病）

级别	总胆固醇	总甘油三酯	高密度脂蛋白	低密度脂蛋白
风险	$\geq 5.7mmol/L$	$\geq 1.70mmol/L$	$\leq 1.04mmol/L$	$\geq 3.37mmol/L$
患病	$\geq 6.2mmol/L$	$\geq 2.26mmol/L$	$\leq 1.04mmol/L$	$\geq 4.16mmol/L$

表 4　疾病标准假设

疾病名称	风险级	患病级
高血压	收缩压 ≥135 或舒张压 ≥85	收缩压 ≥140 或舒张压 ≥90
糖尿病	血糖 ≥6$mmol/L$	血糖 ≥6$mmol/L$
肥胖症	BMI≥24	BMI≥28
高尿酸男	尿酸 ≥380$mmol/L$	尿酸 ≥420$mmol/L$
高尿酸女	尿酸 ≥320$mmol/L$	尿酸 ≥360$mmol/L$

五、 问题分析

5.1 问题一分析

本问题主要是进行数据分析。[1] 首先，依据附件 A3 提出的每条准则及其下细则，对附件 A2 的调查数据进行分析，得到 13 条评价标准，分别得出分析结果。其次，结合 13 条标准，综合分析居民的饮食习惯的合理性。最后得出存在的主要问题。

5.2 问题二分析

本问题首先在问题一的基础上增加了表示生活习惯的指标，形成了饮食和生活习惯指标组，再提取年龄、性别、婚姻状况、文化程度、职业作为个人基本信息情况组，然后对这两组指标进行**典型相关分析**，判断并得到这两组指标之间的相关信息。

5.3 问题三分析

本问题主要分析五种常见慢性病：高血糖、高血压、高血脂、肥胖、高尿酸。首先提取是否患上述五种疾病为定类指标（健康、易患病、患病），然后在上两问的基础上添加吸烟等指标作为影响因素指标，将其与疾病指标作**典型相关分析**，初步得到影响每个慢性病的几个主要因素。得到主要因素后再利用 **XGBOOST-SHAP 模型** [2] 得到常见慢性病（如高血压、糖尿病等）与吸烟、饮酒、饮食习惯、生活习惯、工作性质、运动等因素的关系以及相关程度。

5.4 问题四分析

本问题选择两个指标作为分类依据。首先，依据问题三结果和专业医学知识，将五种慢性病合并为一个指标。其次，依据**卡方检验**的结果将年龄作为另一个划分指标，据

此将所有居民分成九类。最后，分别对不同年龄段的群体重新利用 **XGBOOST-SHAP 模型**分析患病情况与生活饮食等习惯之间的关系，据此给出有利于不同群体身体健康的膳食、运动等方面的建议。

六、 问题一的模型建立与求解

6.1 居民饮食习惯合理性分析

为了评价居民饮食习惯的合理性，依照附件 A3《中国居民膳食指南》提出的八大准则以及调查问卷统计得到的信息，从食物种类和多样性、食物摄入量、营养均衡性等角度定义 13 条指标，如表 5所示。

表 5　饮食习惯评价指标
（按照图 2的顺序排列）

指标名称	指标含义	评价标准
light_wine	酒精摄入量	成人每日摄入 < 15g
healthy_beverage	饮料摄入量	每日喝饮料杯数少于 1 杯
healthy_cooking	在家烹饪频率	至少有一半时间在家吃饭
fresh_vegetables	新鲜蔬菜摄入量	每日摄入 ≥300g
cereal	全谷物摄入量	每日摄入 50~150g
balanced_diet	饮食均衡性	每日膳食包括谷薯类、蔬菜水果、畜禽鱼蛋奶和豆类食物
fresh_fruits	新鲜水果摄入量	每日摄入 200~350g
lpfem	畜禽鱼蛋奶摄入量	平均每日摄入 120~200g
dairy_products	奶制品摄入量	每日摄入 300~500g
beans	大豆制品摄入量	每日摄入 30~50g
healthy_oil	烹饪油摄入量	每日摄入 25~30g
light_salt	食用盐摄入量	成人每日摄入食盐 < 5g
food_diversity	食物多样性	日摄入 12 种以上，周 25 种以上

统计数据对每项指标的满足情况，得到图 2。其中蓝色代表符合指标的部分，红色代表不符合指标的部分。根据指标满足度可以初步得到以下信息：

1. **酒精摄入量**、**饮料摄入量**二者指标满足率很高，超过了 90%，说明了该市居民在“饮”这一部分达到了健康的标准，符合第五条**“控糖限酒”**的准则。
2. **在家烹饪率**这一指标满足率较高，在 80% 左右，说明了该市大部分居民倾向于主要在家饮食，基本符合第七条**“会烹”**的准则。
3. 关于第三条准则**“多吃蔬果、奶类、全谷、大豆”**，该市居民符合程度较低，只有**新鲜蔬菜摄入量**这一指标勉强达到 50%，其余相关的指标如**全谷物摄入量**、**新鲜水果摄入量**、**奶制品摄入量**、**大豆制品摄入量**等满足度基本在 20% 左右，由此可以发现该市居民在健康饮食方面存在较大的问题。
4. 从**食盐摄入量**的指标满足度中得出，该市居民并没有满足第五条**“少盐”**的准则。
5. 关于第一条准则**“食物多样，合理搭配”**，可以发现这一条满足度极低，满足度不足 1%，这是该市居民在饮食习惯方面的关键问题。

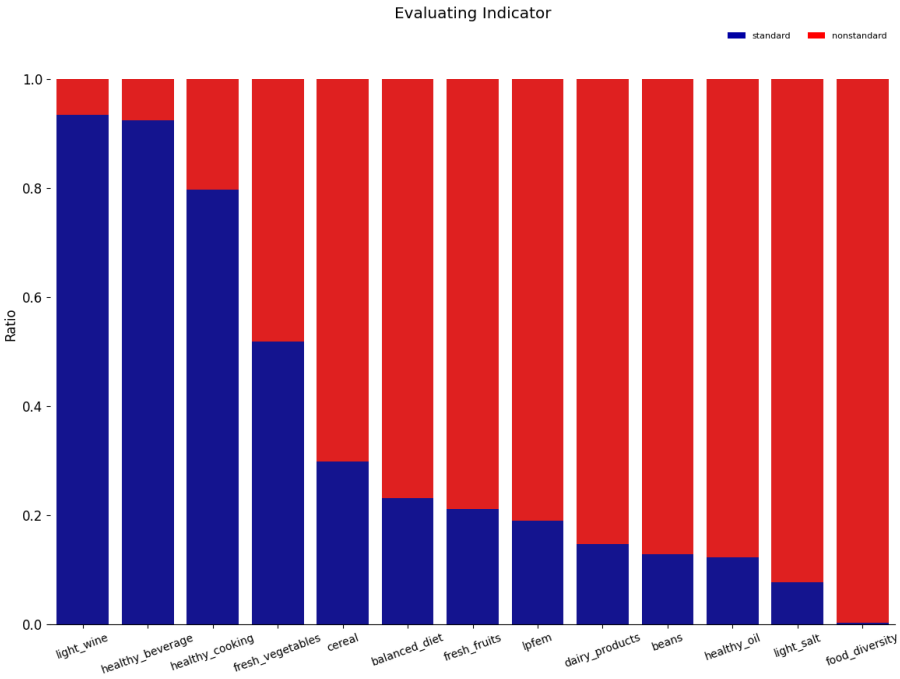
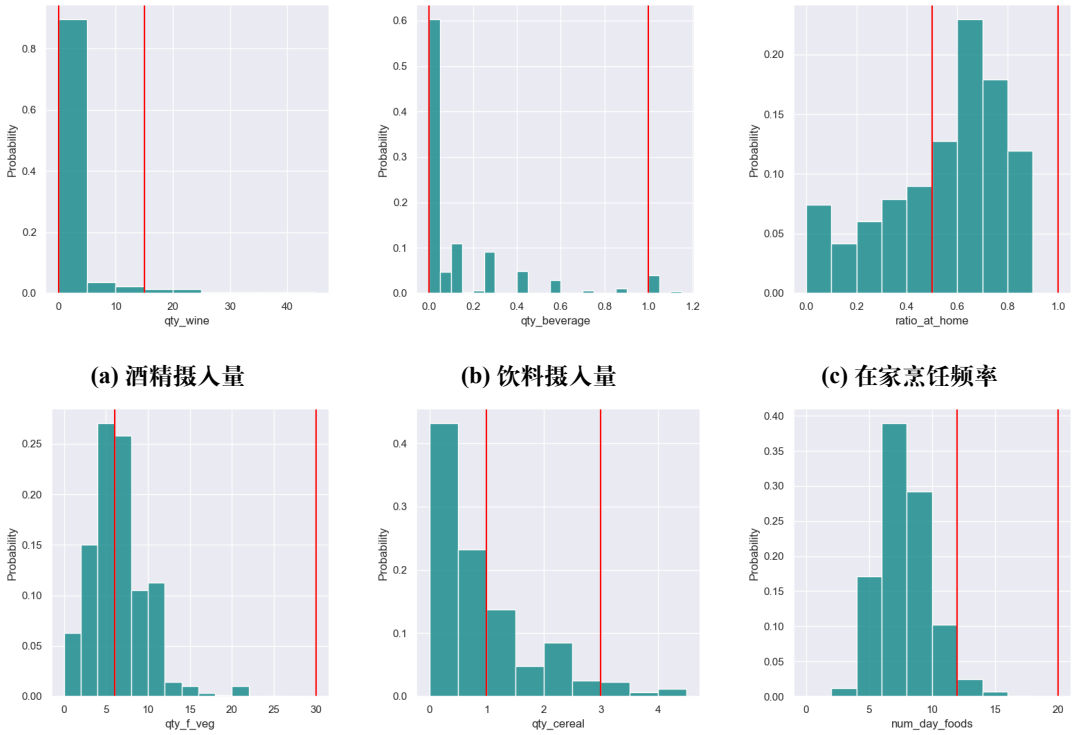


图 2　各指标满足情况

6.2 存在的主要问题

在初步观察和分析 13 组指标后，我们发现由于一些指标的评价标准存在一定的取值范围（例如全谷物摄入量规定每日摄入 50~150g），所以指标的满足与否虽然能在一

定程度上代表居民的饮食习惯健康水平，但无法直观地显示出居民在某一个指标的“过量”或者“不足”。因此，我们为了进一步观察各指标具体的数据分布情况，绘制了指标数据分布的直方图，如图 3所示（此处显示其中 6 个，完整直方图见附录图 11）。



(a) 酒精摄入量 **(b) 饮料摄入量** **(c) 在家烹饪频率**

(d) 新鲜蔬菜摄入量 **(e) 谷物摄入量** **(f) 食物多样性**

图 3　各指标分布直方图

在经过分析、总结和提炼之后，我们发现了该地居民在饮食习惯的几个主要问题。

1. **结构单一**：居民饮食结构单一化严重，荤素搭配、主副搭配严重不合理；
2. **膳食不均**：居民膳食较为不均，全谷物摄入量偏少；
3. **偏食严重**：居民在大豆制品、蔬菜水果与奶制品的摄入量普遍偏少，而畜禽鱼蛋奶肉的摄入量普遍偏高；
4. **喜油喜盐**：食用盐、烹饪油摄入量严重偏高。

七、 问题二的模型建立与求解

7.1 确定算法

本问题要求分析居民生活、饮食习惯和个人基本情况之间是否相关，即研究 {生活习惯，饮食习惯} 和 {年龄、性别、婚姻状况、文化程度、职业} 两组变量之间的线性关系，其中生活饮食习惯可以沿用问题一的思想，定义多项变量进行描述。数据具有变量多、维度高、维度不同的特征。在本题中选用**典型相关分析（CCA）**方法，原因如下：

- 对于数据的多变量性：CCA 能够进行多维度关联分析，可以同时分析多个变量之间的关系，帮助理解数据的多维度关联，发现潜在的关联性和模式。
- 对于数据的高维度性：CCA 能够将多个变量整合到几个典型变量中，从而降低数据的维度，减少分析的复杂性，有助于数据整合。
- 对于数据集维度的差异性：CCA 整合所得典型变量的数量最多与较小数据集中的特征一样多，能够对不同维度的数据集进行分析。

7.2 CCA 模型介绍

CCA（canonical correlation analysis、典型相关分析）模型是利用综合变量对之间的相关关系来反映两组指标之间的整体相关性的多元统计分析方法。它的基本原理是：为了从总体上把握两组指标之间的相关关系，分别在两组变量中提取有代表性的两个综合变量 U1 和 V1（分别为两个变量组中各变量的线性组合），利用这两个综合变量之间的相关关系来反映两组指标之间的整体相关性。

其算法介绍如下：

给定两组向量 x_1 和 x_2，x_1 维度为 p_1 ，x_2 维度为 p_2 ，默认 $\mathrm{p}_1 \leq \mathrm{p}_2$ 。形式化表示如下:

$$\mathrm{x} = \begin{bmatrix} \mathrm{x}_1 \\ \mathrm{x}_2 \end{bmatrix} E[\mathrm{x}] = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \quad \Sigma = \mathrm{Var}(x) = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$$

定义

$$\mathrm{u} = a^T \mathrm{x}_1 \mathrm{v} = b^T \mathrm{x}_2$$

算出 u 和 v 的方差和协方差:

$$\mathrm{Var}(\mathrm{u}) = a^T \Sigma_{11} a \, \mathrm{Var}(\mathrm{v}) = b^T \Sigma_{22} b \, \mathrm{Cov}(\mathrm{u}, \mathrm{v}) = a^T \Sigma_{12} b$$

然后计算出 Corr (u, v) ：

$$\mathrm{Corr}(\mathrm{u}, \mathrm{v}) = \frac{a^T \Sigma_{12} b}{\sqrt{a^T \Sigma_{11} a} \sqrt{b^T \Sigma_{22} b}}$$

利用 Corr (u, v) 可以进一步得到在此典型变量下 x_1 和 x_2 中变量的相关性。

7.3 数据预处理

为了评估居民的生活习惯和饮食习惯，根据调查问卷统计所得的信息，在问题一关于饮食习惯的 13 条指标的基础上再添加 2 条指标，以评价居民生活习惯的合理性，如表8所示。

将数据集划分成三组，第一组仅包含与居民饮食习惯有关的特征，第二组仅包含与居民生活习惯有关的特征，第三组仅包含与年龄、性别、婚姻状况、文化程度、职业等个人基本信息情况相关的特征。

表 6　生活习惯评价指标

指标名称	指标含义
exe_seconds	平均每天体育锻炼时间
work_intensity	工作强度（包括家庭工作）

由于不同变量的尺度不同，为了避免变量尺度影响分析计算过程，增强分析的稳定性，对数据执行**标准化**操作，得到规范变量。

7.4 进行实验

首先，分别计算饮食、生活习惯指标与个人基本信息指标之间的 Pearson 相关系数，观察所有特征之间是否存在相关性，如图 12（见附页）、图 4所示。可以发现，畜禽鱼蛋奶摄入量与奶制品摄入量呈强正相关，这符合人们的直觉认知。

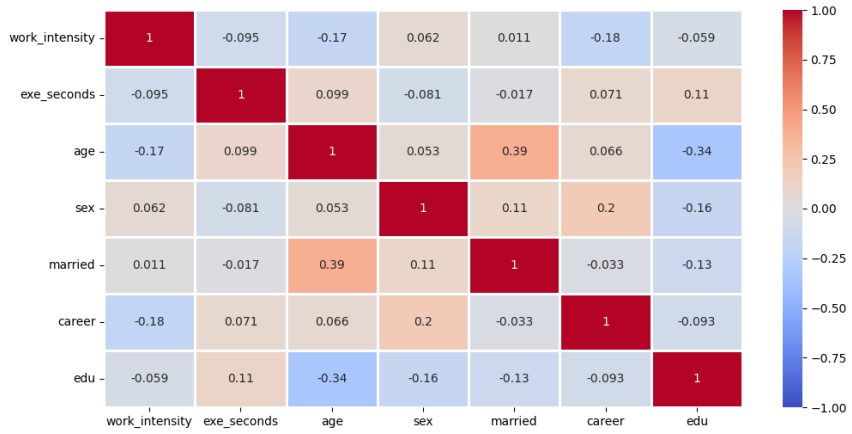


图 4　生活习惯与个人基本指标之间的相关关系

接下来分别分析第一组、第二组数据与第三组数据中变量的相关关系。计算变量之间的**典型变量系数**，得到图 13(见附页）和图 5。典型变量系数表示了每个原始变量在典型变量空间中的投影权重，而变量的投影在一定程度上保留了它们在原始空间中的关系。如果两个变量在典型变量空间中的投影方向相似且权重较大，那么它们在原始空间中的相关性也较大。观察热力图可以直观地得出，一部分饮食习惯指标与个人基本信息相关性显著，另一部分则不存在显著的相关性。

具体可以得出以下结论：

1. 摄入食物的多样性与性别、文化程度存在较弱的正相关性。

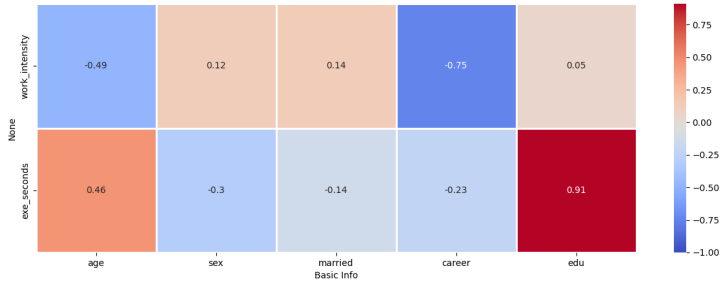


图 5　生活习惯与个人基本信息的关系

2. 奶制品摄入量与性别存在较强的正相关性，与文化程度存在较弱的正相关性。

3. 畜禽鱼蛋奶摄入量与性别存在较强的负相关性，与文化程度存在较弱的负相关性。

4. 酒精摄入量与性别存在较弱的负相关性。

5. 在家就餐比例与年龄存在较强正相关性，与性别、职业存在较弱正相关性，与文化程度存在较强负相关性。

6. 居民的工作强度与职业存在较强的负相关性, 与年龄存在较弱的负相关性。

7. 居民的体育锻炼情况与文化程度存在较强的正相关性. 与年龄存在较弱的正相关性。

8. 剩余指标之间均不存在显著的相关性。

总结可以得出，**该市居民生活饮食习惯与婚姻状况不存在显著相关性，饮食习惯与文化程度、性别、年龄相关性较为显著，生活习惯与年龄、性别、职业和文化程度相关性较为显著**。其中，高油高盐是普遍存在的问题，文化程度高的人群饮食、生活习惯更好，女性的饮食习惯更符合健康的标准。

7.5 结果合理性分析

下面结合生活经验和过往的研究成果对结果进行合理性分析。

对于结论 1，可以解释为文化程度越高往往意味着对健康有更紧迫的追求和更科学充分的理解，也就越能有意识地摄入多样性的食物。

对于结论 2，分析结果与其他文献 [4] 结果相吻合。女性较男性摄入更多的奶制品，可能的原因有媒体对牛奶美白美容功效的过度宣传；文化程度高的群体摄入奶制品更多，可能是由于他们对知识的接受能力更强，对身体健康更为关注，更容易理解和接受科学健康的资讯，也就更可能认识到摄入适量奶制品的益处与重要性。[4]

对于结论 3，肉禽鱼蛋奶均属于动物性食物，它们不仅含有丰富的蛋白质、脂肪、无机盐和维生素，而且蛋白质的质量高，属优质蛋白。首先，性别方面的关联表明男性和女性在肉禽鱼蛋奶摄入方面可能存在着不同的偏好或行为模式，这可能是由于男性和

女性对于蛋白质等营养需求的差异，以及饮食文化和社会角色的影响；其次，与文化程度的关联可能是多种因素复杂交织的结果，例如文化程度较高往往意味着更强烈的健康意识和更多元的信息导向，他们可能将素食或植物性饮食视为更环保、更健康、更可持续的生活方式，这使得他们更多地倾向于选择植物性食物，进而减少了肉禽鱼蛋奶的摄入。

对于结论 4，给出两种可能的解释。从社会文化因素的角度来说，男性往往更多地参与到社交聚会、商务活动等与饮酒相关的场合，面临的饮酒压力也更高。从生物学因素的角度来说，由于男性平均体重更大、身体含水量更高、平均体脂率更低，在摄入等量酒精后，男性身体内的酒精浓度往往更低。

对于结论 5，容易解释为年长的人工作时间减少，且更注重健康和饮食，可能有更多的时间参与家庭烹饪，尤其是有子女的情况下；男性和女性在职业和生活方式方面的差异，以及不同工作的性质，导致家庭就餐比例的不同；文化程度较高的人可能有更多的职业和社交活动，导致在家用餐比例降低，且他们的收入水平往往更高，能够承担更多外出就餐的费用。

对于结论 6 和结论 7，职业对工作强度的影响是显然的，年龄增大则意味着精力的下降，从事的工作强度往往也随之降低。关于体育锻炼情况，可能的一种解释是，体育锻炼意识与人的受教育程度成正相关关系 [3]，且高学历群体的工作往往缺少身体活动，因此文化程度越高的人群越倾向于投入更多的空闲时间到体育锻炼中。而年龄的增加意味着空闲时间的增多，健康意识也往往随之增强，投入身体锻炼的时间更多。

综上所述，模型分析所得结果合理。

八、 问题三的模型建立与求解

8.1 指标确定

本题要求分析常见慢性病和生活、饮食习惯，饮酒、吸烟情况等因素的关系。首先，根据已有的数据信息，提取了五个关于是否患有慢性病的定类指标（0：健康、1：易患病、2：患病），具体患病情况确定参考 4.4 节慢性病评判标准。指标内容如表 7 所示：

除此之外，在该问中添加了吸烟情况作为新指标，形成 16 条影响因素指标。

8.2 CCA 初步分析

首先，我们沿用问题二的**典型相关分析（CCA）**方法，对五项慢性病指标和 16 条影响因素指标的相关关系进行初步分析，得到图 6。

表 7　慢性病评价指标

指标名称	指标含义
hypertension	患高血压情况
diabetes	患高血糖情况
obesity	患肥胖情况
high_uric_acid	患高尿酸情况
hyperlipidemia	患高血脂情况

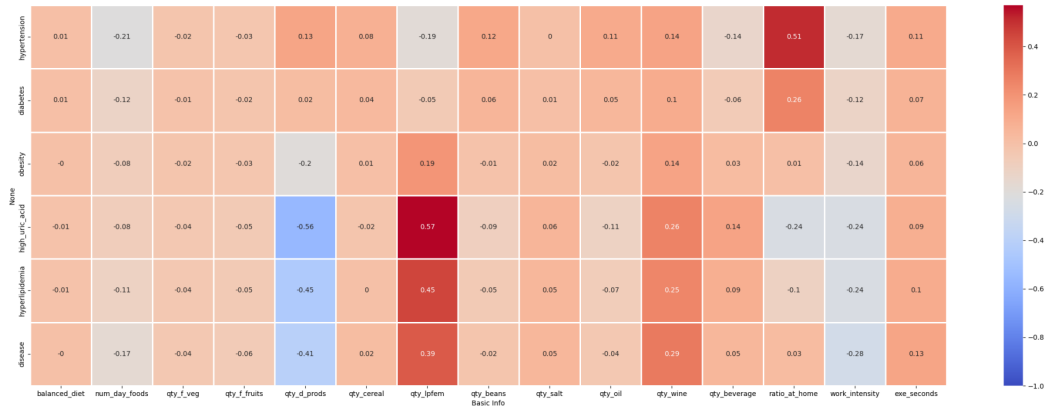


图 6　常见慢性病与吸烟、饮酒、饮食习惯、生活习惯、工作性质、运动等因素的关系

观察热力图，可得到以下初步结论：

1. 患高血压与在家就餐的频率呈正相关。
2. 患高尿酸与奶制品摄入量呈负相关，与畜禽鱼蛋奶摄入量呈正相关。
3. 患高血脂与奶制品摄入量呈负相关，与畜禽鱼蛋奶摄入量呈正相关。

对于以上结论，我们给出以下可能的解释：

1. 从饮食结构和成分角度来说，在家吃饭时，人们往往更容易接触到高盐、高油、高糖的食物，这些食物的摄入过多可能导致体内钠水平升高，增加血压的风险；从饮食习惯和控制角度来说，在家吃饭时，人们往往更容易大量摄入高热量、高脂肪的食物，如油炸食品、甜点、碳酸饮料等，长期以来可能导致体重增加，进而增加高血压的发生率；从运动和生活方式来说，在家吃饭次数增多可能导致身体活动减少，缺乏足够的运动和身体活动会增加肥胖、心血管疾病和高血压的风险。

2. 尿酸是人体嘌呤代谢的产物。关于奶制品摄入，由于奶制品通常属于低嘌呤食物，因此摄入奶制品可能减少体内嘌呤的积累，从而影响尿酸水平；奶制品含有的

一些成分，如乳清蛋白，有助于尿酸的排泄，从而可能导致患高尿酸的风险降低。关于畜禽鱼蛋奶摄入，畜禽鱼蛋奶摄入较多的人群可能同时摄入其他类食物较少，导致饮食不均衡，不利于平衡尿酸水平。

3. 奶制品中的脂肪含量相对较低，特别是低脂或脱脂的奶制品，而畜禽鱼蛋奶中的脂肪含量较高，不恰当的摄入可能会影响高血脂的风险。

8.3 XGBOOST-SHAP 模型

在利用 CCA 进行初步分析之后，我们采用了**XGBOOST-SHAP 模型**进一步深入分析常见慢性病和生活、饮食习惯等因素的关系和相关程度。

8.3.1 模型介绍

SHAP 的名称来源于 SHapley Additive exPlanation。Shapley value 起源于合作博弈论。SHAP 是由 Shapley value 启发的可加性解释模型。对于每个预测样本, 模型都产生一个预测值, SHAP value 就是该样本中每个特征所分配到的数值。假设第 i 个样本为 x_i, 第 i 个样本的第 j 个特征为 $x_{i,j}$, 模型对第 i 个样本的预测值为 y_i, 整个模型的基线（通常是所有样本的目标变量的均值）为 y_{base}, 那么 SHAP value 服从以下等式。

$$y_i = y_{\text{base}} + f(x_{i,1}) + f(x_{i,2}) + \cdots + f(x_{i,k})$$

其中 $f(x_{i,1})$ 为 $x_{i,j}$ 的 SHAP 值。直观上看, $f(x_{i,1})$ 就是第 i 个样本中第 1 个特征对最终预测值 y_i 的贡献值, 当 $f(x_{i,1}) > 0$, 说明该特征提升了预测值, 也正向作用; 反之, 说明该特征使得预测值降低, 有反作用。SHAP value 最大的优势是 SHAP 能对于反映出每一个样本中的特征的影响力, 而且还表现出影响的正负性。

8.3.2 分析结果

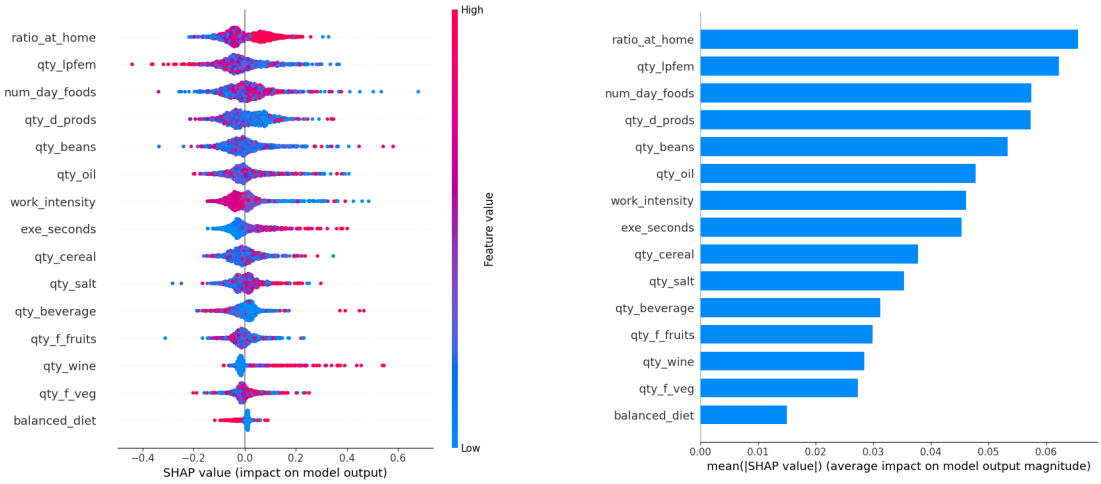
我们对五种疾病分别采用 XGBOSST-SHAP 模型进行分析，得到了不同疾病的主要影响因素。我们以高血压和高尿酸为例对模型结果进行分析，其余疾病的模型分析结果图放在附录中。

不同指标对高血压的影响情况如图 7所示：

在图 19a中，每一行代表一个特征，横坐标为 SHAP 值。一个点代表一个样本，颜色越红说明特征本身数值越大，颜色越蓝说明特征本身数值越小。我们从中可以看出酒精摄入量 (qty_wine) 中红色点集中在 SHAP 大于 0 的区域，证明喝酒多的人容易患高血压。在图 19c中，每一行表示特征对目标变量影响程度的绝对值的均值。可以看出在家就餐的频率，摄入畜禽鱼蛋奶的量等因素对是否会患高血压的总体影响较大。

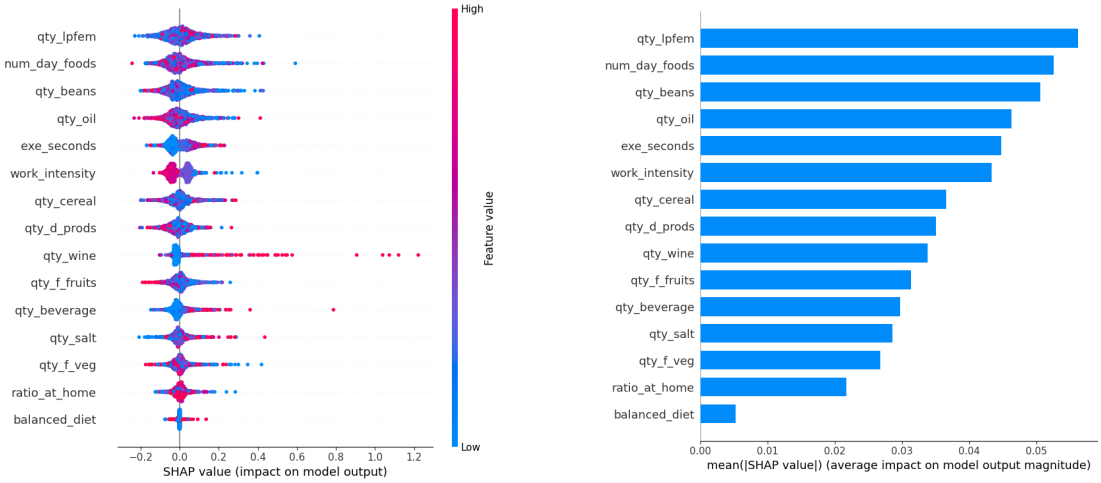
不同指标对高尿酸的影响情况如图 8所示：

可以从图 8中得到以下几个结论：



(a) (b)

图 7　高血压受指标影响情况



(a) (b)

图 8　高尿酸受指标影响情况

- 酒精摄入量多，奶制品摄入量多，工作强度不高的人容易患高尿酸，可以在左图中看到这几个指标的红蓝色具有明显的分布。
- 谷物摄入量，每天吃的食物种类等因素对高尿酸的综合影响较大（这些指标位于图的上侧），但量的多少不具有明显的影响边界（左图中显示红蓝色点融合在一起）。

对高血糖、肥胖、高血脂等疾病影响因素的分析图见附录，分析结果见表 8：

表 8　慢性病影响因素分析表

慢性病	分析结果
高血糖	酒精摄入量多、工作强度低的人群容易患高血糖 豆类摄食量、每天食物种类等对高血糖的综合影响较大，但没有清晰的影响边界
肥胖	水产品摄入少，谷物摄入多，酒精摄入多，饮食不均衡的人群容易患肥胖 水产品，谷物等摄入对肥胖慢性病的影响都较大
高血脂	谷物摄入少，肉食蛋奶摄入多，酒精摄入多的人群容易患高血脂 谷物摄入对高血脂的影响最大

九、 问题四的模型建立与求解

9.1 分类

9.1.1 分类标准

由第三问的分析可以看出，五种疾病的影响因素存在较大的共性，如谷物摄食量，酒精摄食量等。同时参考相关资料得到，这五种疾病都属于代谢异常疾病。因此，我们把这五种疾病指标统一为一项指标，用 0、1、2 分别表示健康、易患病、患病。

除此之外，由于不同年龄段的人的体质、生活习惯有较大差异，我们将年龄也作为一项分类标准。其中 44 岁以下为青年人，45-59 岁为中年人，60 岁以上为老年人。

9.1.2 分类依据

我们对患病指标和年龄指标进行卡方检验。

对于 rxc 的列联表，若其总样本数大于 40 且每个类别的理论频数大于等于 5 可用卡方检验。

$$H_0 : p_{ij} = p_{i\cdot} \cdot p_{\cdot j}, \quad i = 1, ..., r, j = 1, ..., c$$

检验统计量为:

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(n_{ij} - n\hat{p}_{ij})^2}{n\hat{p}_{ij}}$$

在原假设 H_0 成立时上式近似服从自由度为 (r − 1)(c − 1) 的 χ^2 分布。其中各 $\hat{p}_{ij}$ 是在 H_0 成立下得到的 $\hat{p}_{ij}$ 的最大似然估计，其表达式为:

$$\hat{p}_{ij} = \hat{p}_{i\cdot} \cdot \hat{p}_{j} = \frac{n_{i\cdot}}{n} \cdot \frac{n_{j}}{n}$$

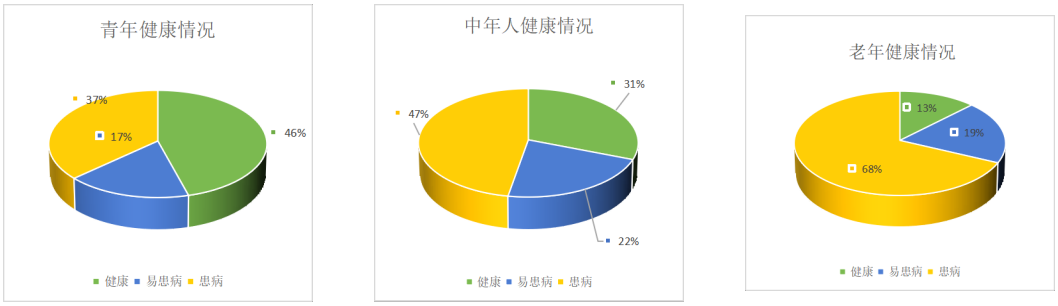
对给定的显著性水平 α，检验的拒绝域为 $W = \chi^2$

对于本问数据，统计得到列联表如下：

	青年人	中年人	老年人	All
健康	888	341	714	1943
易患病	1193	850	1826	3869
患病	253	393	1378	2024
All	2334	1584	3918	7836

计算得到卡方值为 589.79，p 值为 0，拒绝原假设，证明年龄对患病情况有显著性影响。

为了更直观地呈现年龄对患病情况的显著性影响，我们对各年龄段的患病情况制作了三维饼图，如图 9所示。（各年龄段的各种疾病的患病情况饼图见附录图 17）



(a) (b) (c)

图 9　年龄对患病情况的显著性影响

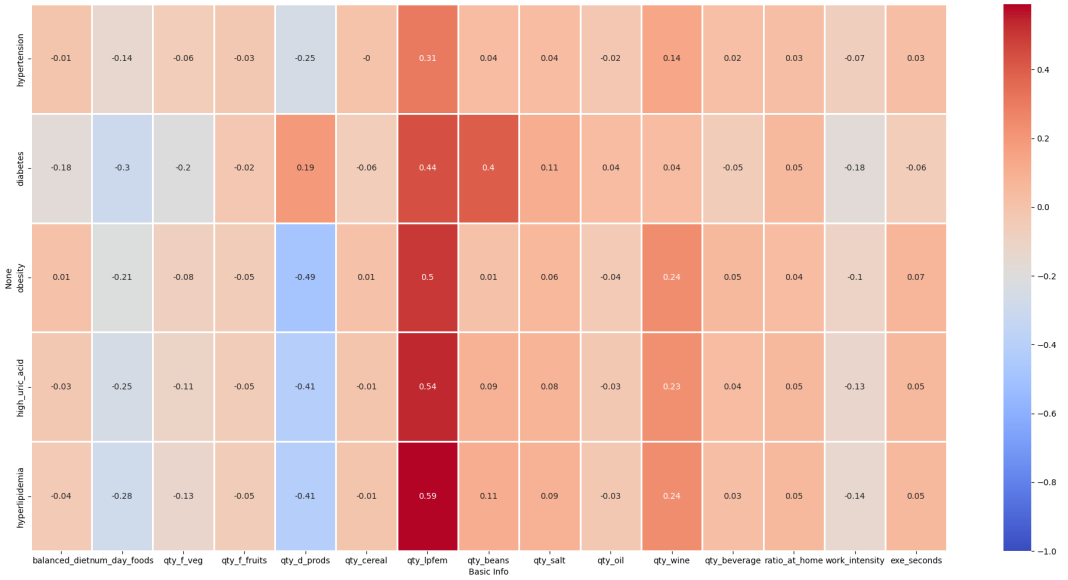
9.2 膳食与运动建议

在上述分类的基础上，我们对三种不同年龄段群体重新进行了第三问的分析，利用 CCA 和 XGBOOST-SHAP 模型分析不同人群的患病情况与生活习惯等的关系。结果如下（其中中年人、老年人的模型结果图见附录）：

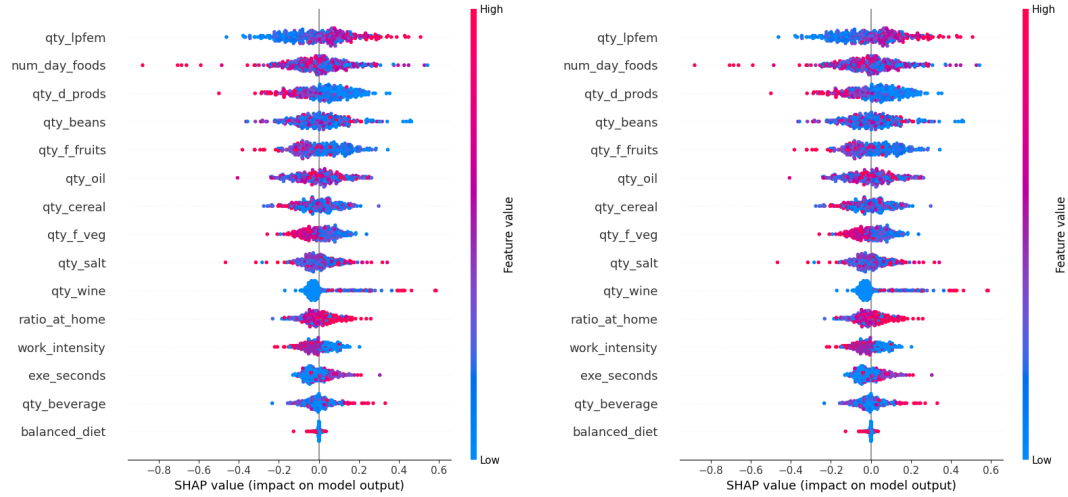
由图图 10、图图 18、图图 19可以看出，对于所有年龄段的人而言，共同的特点是是否患慢性病和全谷类、畜禽鱼蛋奶摄入量相关性比较大，因此对所有年龄段的人的健康建议为多吃全谷物食物，控制吃畜禽鱼蛋奶类食品的量。

对于青年人而言，全谷类、畜禽鱼蛋奶摄入量对健康的影响在整个年龄段最为显著，因此青年人应该特别重视好的饮食习惯的养成。

对于中年人而言，是否患慢性病和饮酒情况的相关性比较大，五种疾病均呈现正相关。因此可以证明饮酒过量给中年人健康带来的危害最大。所以给中年人的建议要着重强调少饮酒，控制酒精摄入量。



(a)



(b) **(c)**

图 10　青年人患病的影响因素

对于老年人而言，是否患慢性病和工作强度的相关性较大。从事重体力活动的老年人反而不容易患慢性病，因此给老年人的建议是平常多运动，不要久坐，这样可以降低患慢性病的概率。

十、 模型评价与改进

10.1 优点

1. **对于数据规模大、维度高的特点，选择合适的处理方式和模型。**
   问题最初选择建立 PERSON 类管理数据，
2. **结合专业营养学、疾病学知识和事实经验分析结果，更具有科学性和合理性。**
   本问题具有一定营养学方面的专业性，为了更合理地进行分析和评价，在数学模型分析的基础上，还结合一定的营养学、疾病学知识对所得结果进行分析，能够给出更合理解释，也有利于对结论进行验证和优化。

10.2 缺点

1. **模型没有全面考虑到问卷中的所有数据**
   本模型对于问卷中家族是否患病等指标并没有将其纳入，不够全面充分，还能进一步进行处理和分析。
2. **模型在分类时采用的评判标准比较单一**
   在最后对人群的分类时，本模型主要依靠年龄进行分类，按照患病程度和指标的关系给出建议。这样的分类方式还有待优化

参考文献

[1] Hongsheng Chen, Ye Liu, Zhenjun Zhu, and Zhigang Li. Does where you live matter to your health? investigating factors that influence the self-rated health of urban and rural chinese residents: evidence drawn from chinese general social survey data. Health and quality of life outcomes, 15(1):1–11, 2017.

[2] De-Cheng Feng, Wen-Jie Wang, Sujith Mangalathu, and Ertugrul Taciroglu. Interpretable xgboost-shap machine-learning model for shear strength prediction of squat rc walls. Journal of Structural Engineering, 147(11):04021173, 2021.

[3] 孟兵林 王海涛杨雯茜, 杨光. 不同学历的 20 岁及以上人群体育锻炼状况研究. 当代体育科技, 6(117-118), 2016.

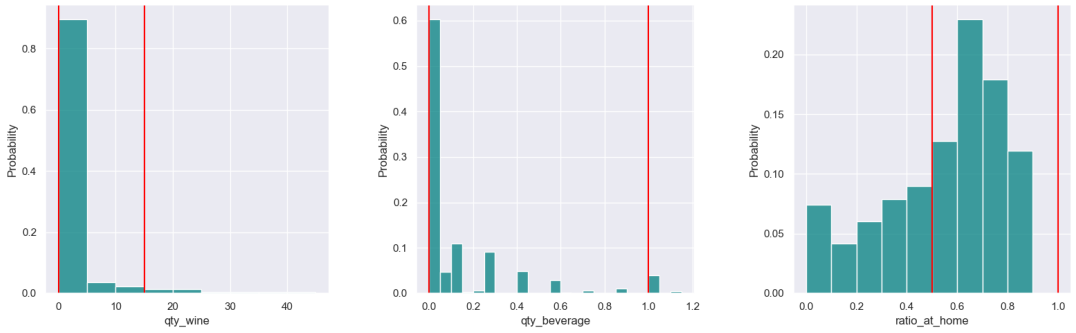
[4] 冯维杰朱凯星梁佳志张弛庄晓霞 郭宁晓邱泉, 栾玉明. 广州某区 15 岁以上人群奶制品摄入及影响因素. 现代预防医学, 41(2532-2534), 2014.

附录 A　无效数据处理表格

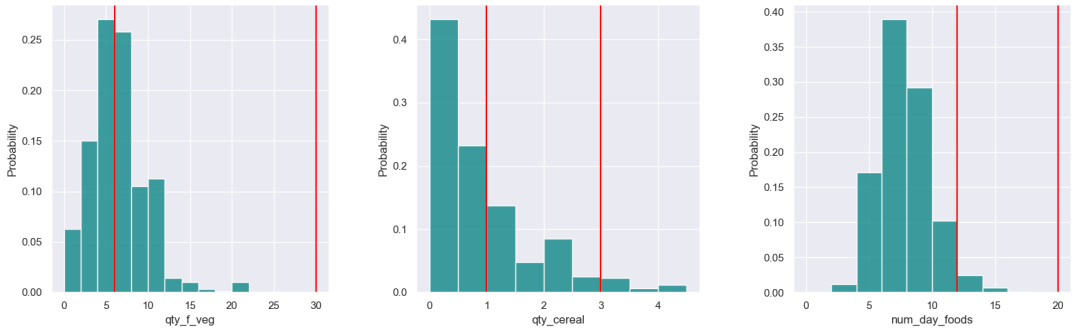
表 9　无效数据处理

所属类别	内容	问题 ID	处理方式
BASIC	部分人的 ID 遗失	17836 等	保持原始顺序下补充
BASIC	未说明民族情况	10232 等	在统计时标记为 None
BASIC	婚姻状况未填写	15780 等	在统计时标记为 None
BASIC	职业状况未填写	15780 等	在统计时标记为 None
DRINK	开始饮酒未填写	10070 等	默认记为 99，记不清
MEALS	用餐人数未填写	10005 等	用假设的 2.62 来代替
FOODS	食用频率多填写	10097 等	以频率最高的为标准
FOODS	食用但频率未填	10101 等	在统计时标记为 None
FOODS	食用量没有填写	10684 等	默认记为 0，不食用
ACTIVITY	家务活动未填写	10082 等	默认为不做家务活动
ACTIVITY	锻炼频率未填写	10093 等	在统计时标记为 None
ACTIVITY	锻炼时间未填写	10019 等	在统计时标记为 None
BODY	身高体重未填写	13715 等	在统计时标记为 None

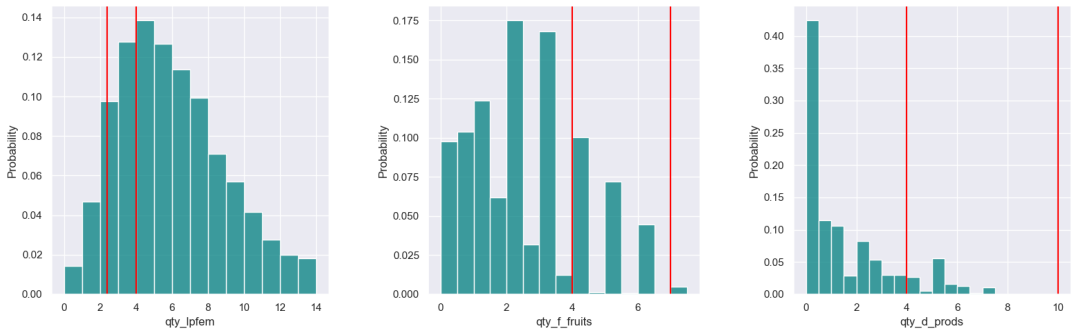
附录 B　指标数据分布直方图



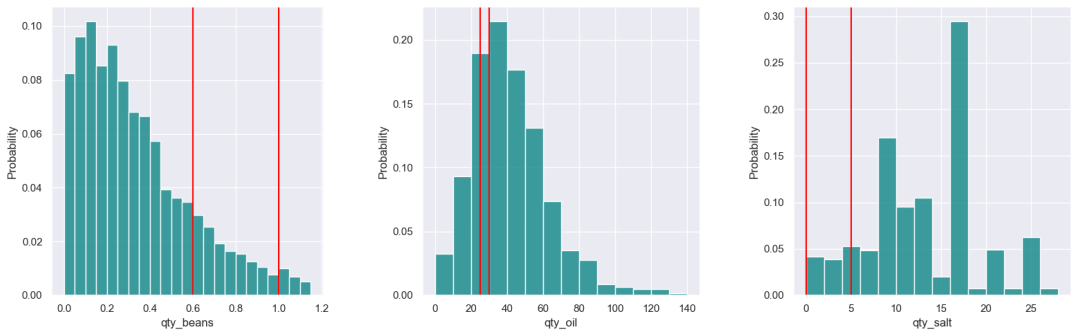
(a) 酒精摄入量　(b) 饮料摄入量　(c) 在家烹饪频率



(d) 新鲜蔬菜摄入量　(e) 谷物摄入量　(f) 食物多样性



(g) 畜禽鱼蛋奶摄入量　(h) 新鲜水果摄入量　(i) 奶制品摄入量



(j) 大豆制品摄入量　(k) 烹饪油摄入量　(l) 食用盐摄入量

图 11　各指标分布直方图

附录 C　问题二典型相关系数分析结果图

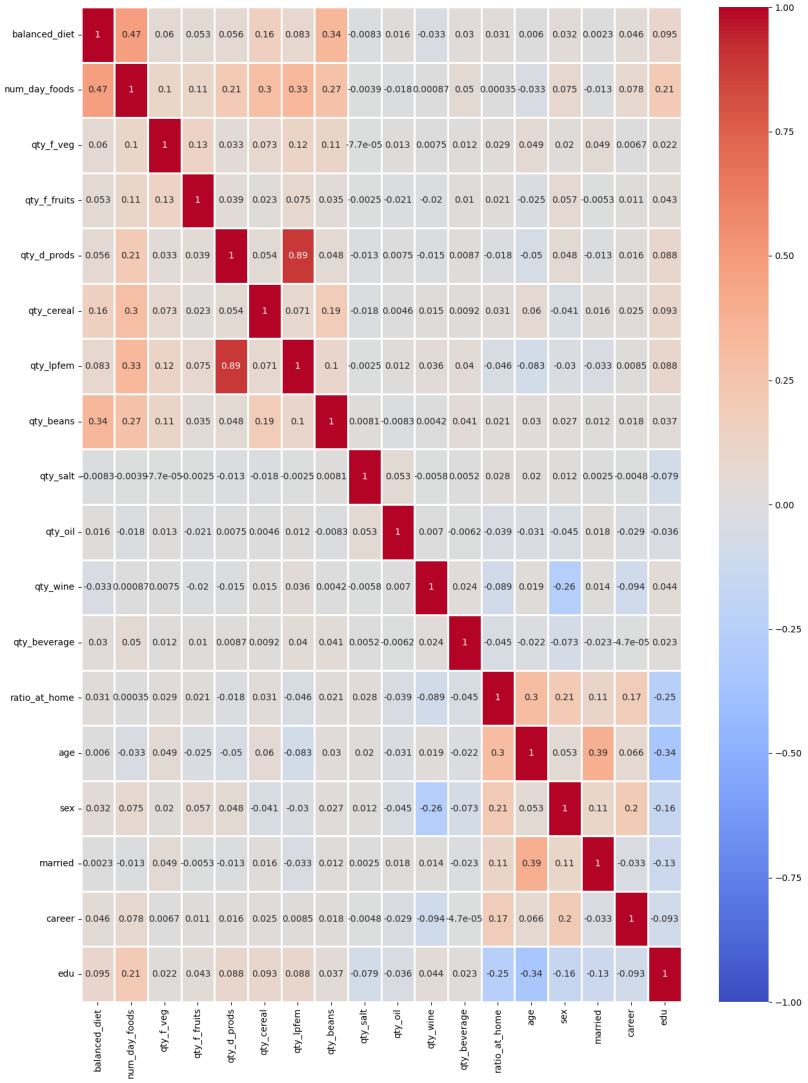


图 12　饮食习惯与个人基本指标之间的相关关系

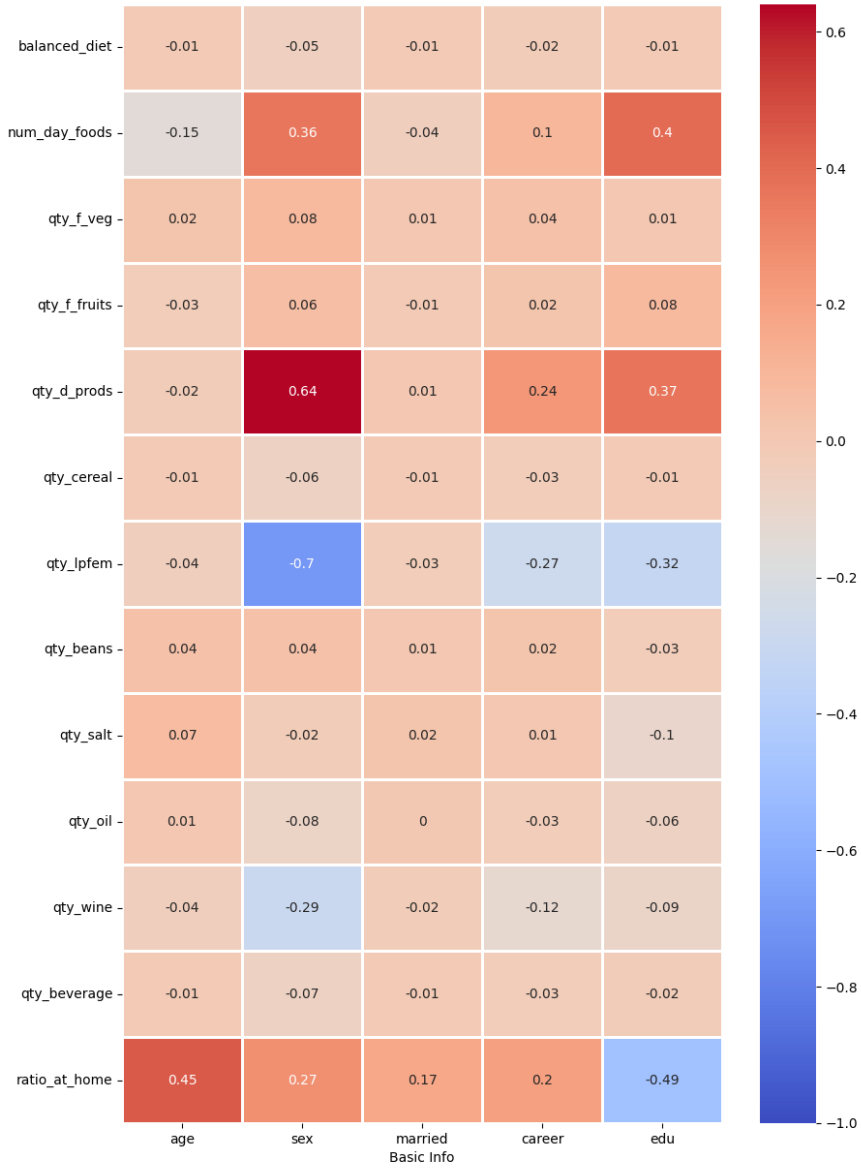
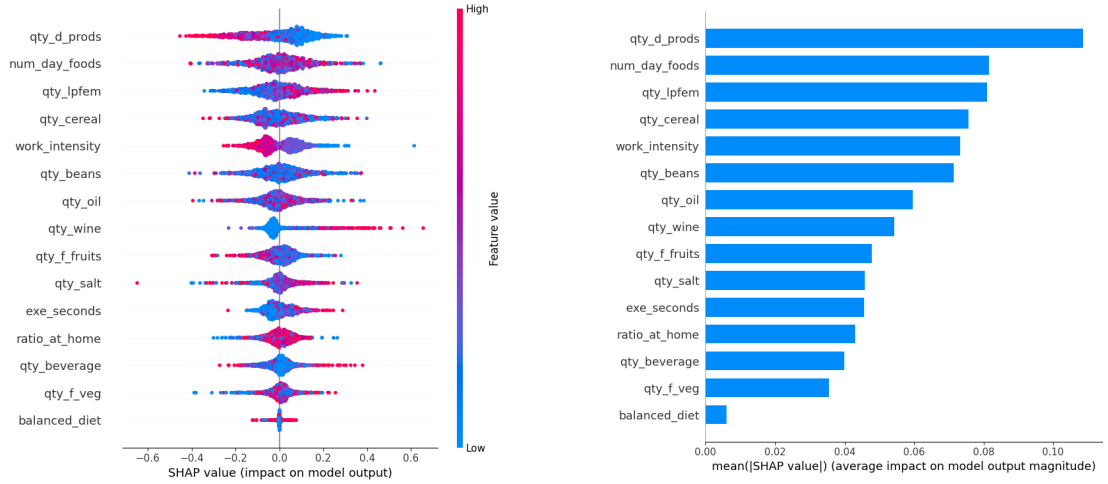


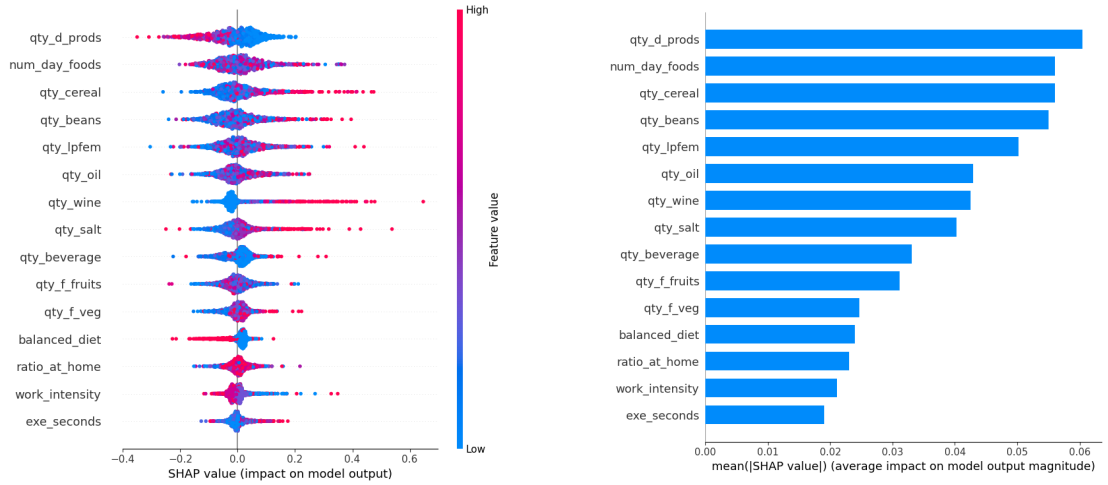
图 13　饮食习惯与个人基本信息的关系

附录 D　问题三其余疾病分析结果图



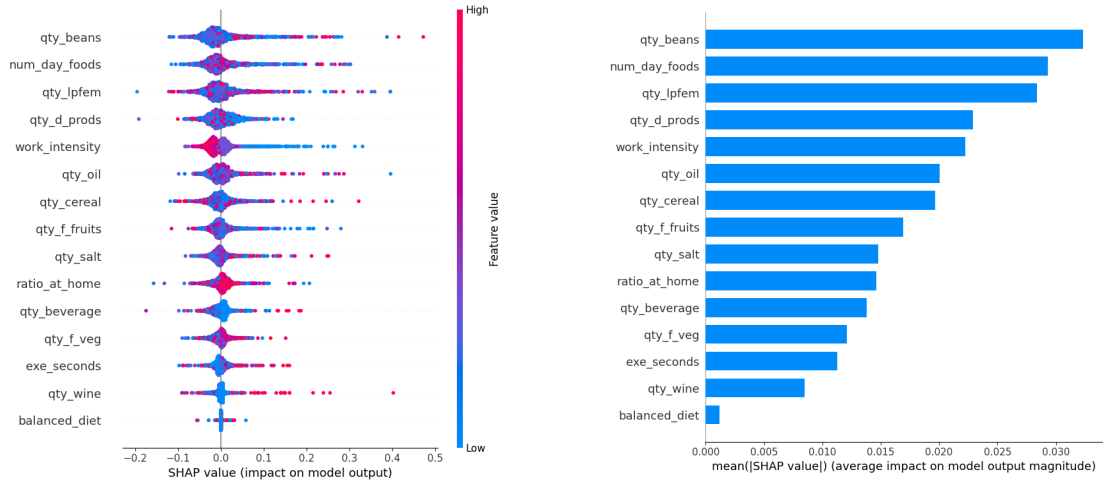
(a)　　(b)

图 14　高血脂受指标影响情况



(a)　　(b)

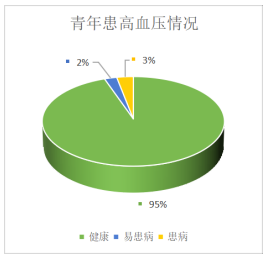
图 15　肥胖受指标影响情况



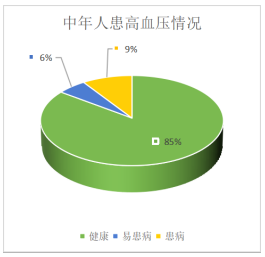
(a) **(b)**

图 16　高血糖受指标影响情况

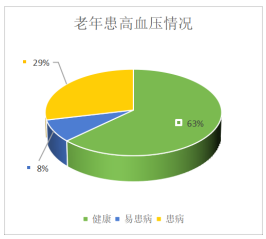
附录 E　问题四各年龄段各种疾病患病情况



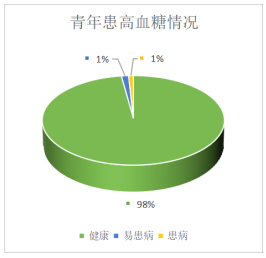
(a) 青年-高血压



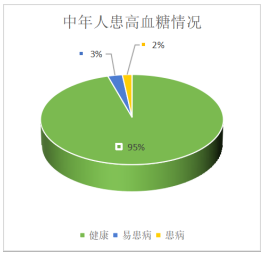
(b) 中年-高血压



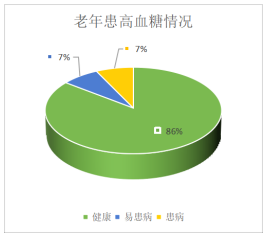
(c) 老年-高血压



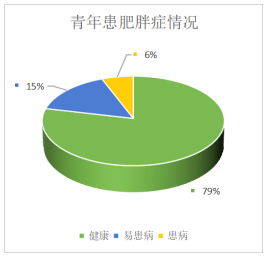
(d) 青年-高血糖



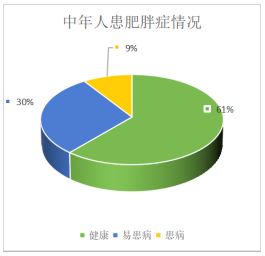
(e) 青年-高血糖



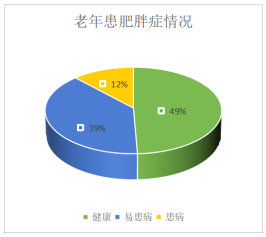
(f) 老年-高血糖



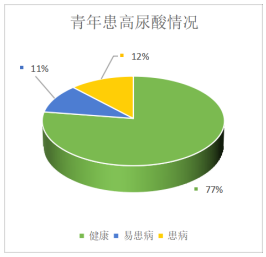
(g) 青年-肥胖症



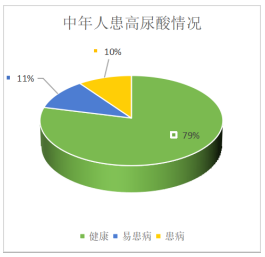
(h) 中年-肥胖症



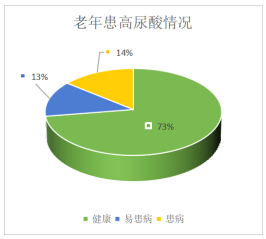
(i) 老年-肥胖症



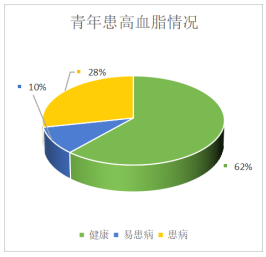
(j) 青年-高尿酸



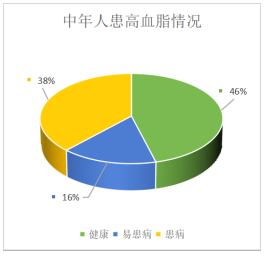
(k) 中年-高尿酸



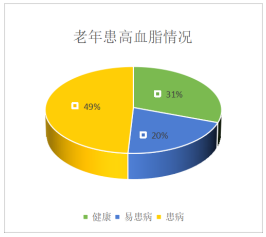
(l) 老年-高尿酸



(m) 青年-高血脂



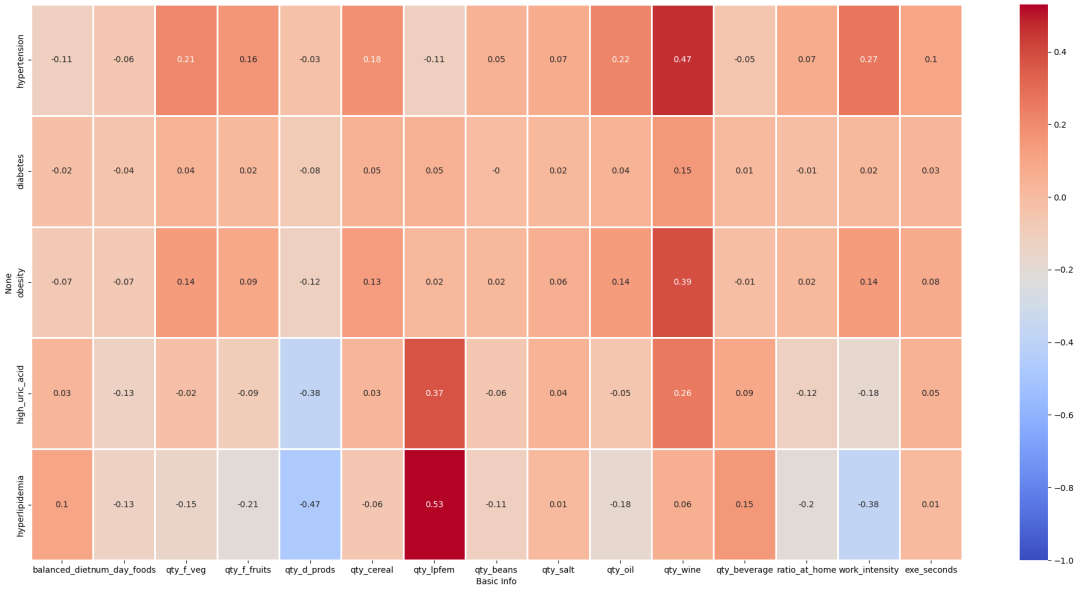
(n) 中年-高血脂



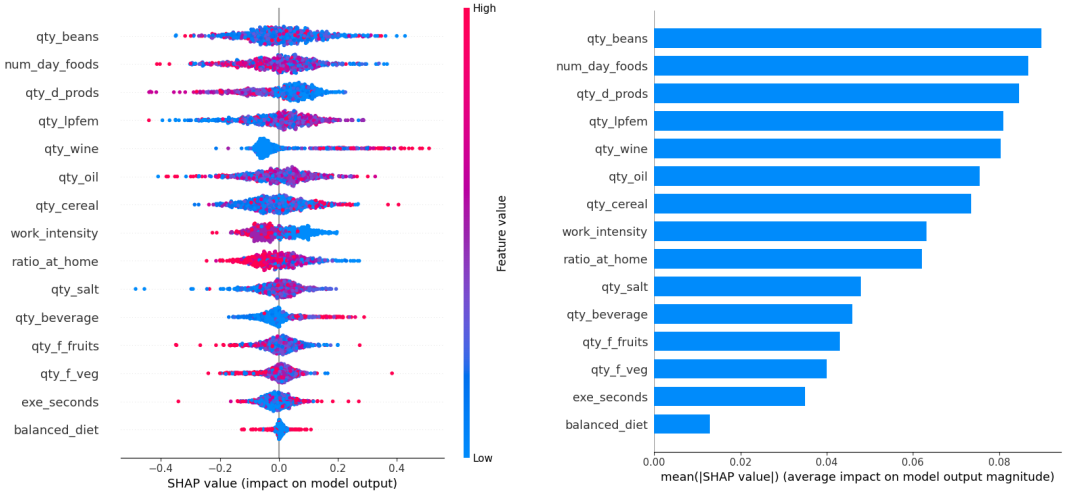
(o) 老年-高血脂

图 17　年龄对患各种慢性病的影响

附录 F　问题四各年龄段患病影响因素分析图

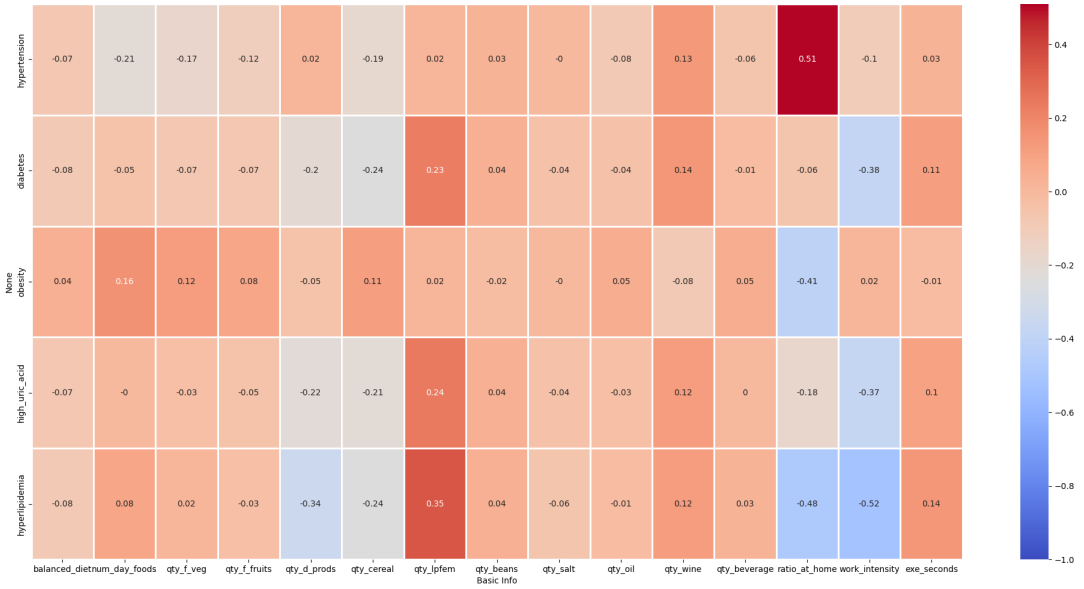


(a)

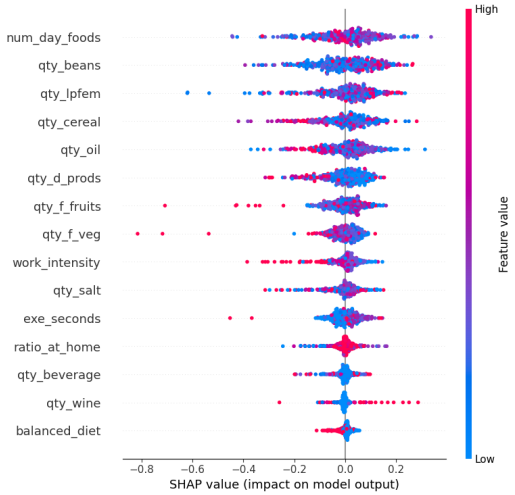


(b)　　**(c)**

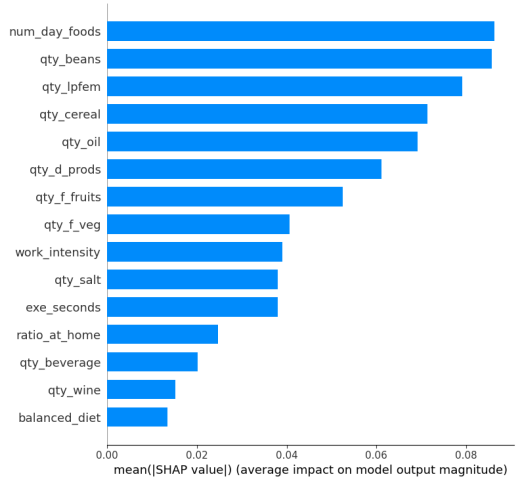
图 18　中年人患病的影响因素



(a)



(b)



(c)

图 19　老年人患病的影响因素

附录 G　代码

这里只列举出部分代码，完整代码仓库在 https://github.com/heatingma/SJTU-2023C

7.1　数据处理代码

Listing 1: Person 类

```

# Person类记录个体的数据以及数据的处理
class Person:
    def __init__(self, data):
        self.basic_info = BASIC(*data[0:8])
        self.smoke_info = SMOKE(*data[8:14])
        self.drink_info = DRINK(data[14], data[15], *[WINE(data[i], data[i+1],
            data[i+2]) for i in range(16, 31, 3)])
        self.meals_info = MEALS(*[MEAL(data[i], data[i+1], data[i+2], data[i+3],
            data[i+4], data[i+5], data[i+6]) for i in range(31, 52, 7)])
        self.foods_info = FOODS(*[FOOD(data[i], data[i+1], data[i+2], data[i+3],
            data[i+4]) for i in range(52, 187, 5)], *data[187:194])
        self.activity_info = ACTIVITY(*data[194:199])
        self.health_info = HEALTH(*[DISEASE(data[i], data[i+1], data[i+2],
            data[i+3], data[i+4],data[i+5], data[i+6]) for i in range(199, 213,
            7)], *data[213:221])
        self.body_info = BODY(*data[221:234])
        self.data_process()
        self.cal_guideline()

    def data_process(self):
        ...

    def cal_guideline(self):
        self.evaluate_info = EVALUATE()
        self.evaluate_info.add_evaluate([...])
        self.evaluate_info.add_qty([...])

    def __repr__(self):
        message = "basic_info, smoke_info, drink_info, meals_info, foods_info, "
        message += "activity_info, health_info, body_info, evaluate_info"
        return f"{self.__class__.__name__}({message})"
```

Listing 2: Persons 类

```
# Persons类记录全体，其中person_dict的value是Person类的对象
class Persons:
    def __init__(self):
        self.person_dict = dict()
        self.message = "person_dict"

    def add_person(self, person: Person):
        self.person_dict[person.basic_info.id] = person # 添加个体

    def statistics(self):
        """
        统计分析
        """
        attrs = self.person_dict[10001].evaluate_info.evaluate_dict.keys()
        self.evaluate_ratio = list()
        self.evaluate_info = list()
        for attr in attrs:
            self._statistics(attr)
            self.evaluate_ratio.append(getattr(self, attr).get_ratio())
            self.evaluate_info.append(getattr(self, attr).get_info())
        self.evaluate_ratio = np.array(self.evaluate_ratio)
        self.evaluate_info = np.array(self.evaluate_info)
        self.evaluate_ratio =
            self.evaluate_ratio[np.argsort(-self.evaluate_ratio[:,
            2].astype(float))]
        self.evaluate_info =
            self.evaluate_info[np.argsort(-self.evaluate_info[:, 3].astype(int))]

    def _statistics(self, name="balanced_diet"):
        total = len(self.person_dict)
        effective = 0
        meet = 0
        data_list = list()
        for person in self.person_dict.values():
            evaluate_info = getattr(person, "evaluate_info")
            evaluate_dict = getattr(evaluate_info, "evaluate_dict")
            data_list.append(evaluate_dict[name])
            if evaluate_dict[name] is not None:
                effective += 1
                meet += int(evaluate_dict[name])
        data_list = np.array(data_list)
        setattr(self, name, STATISTICS(name, data_list, total, effective,
            meet=meet))
        self.message += (", " + name)
```

```
42
43  def get_dataframe(self):
44      """
45      生成csv文件
46      """
47      person_data = pd.DataFrame()
48      person_data_young = pd.DataFrame()
49      person_data_mid = pd.DataFrame()
50      person_data_old = pd.DataFrame()
51
52      for person in self.person_dict.values():
53          append_data = getattr(person.evaluate_info, "evaluate_dict")
54          append_data.update(getattr(person.evaluate_info, "qty_dict"))
55          person_data = person_data._append(append_data, ignore_index = True)
56          if person.basic_info.age_group == 0:
57              person_data_young = person_data_young._append(append_data,
                    ignore_index = True)
58          elif person.basic_info.age_group == 1:
59              person_data_mid = person_data_mid._append(append_data,
                    ignore_index = True)
60          else:
61              person_data_old = person_data_old._append(append_data,
                    ignore_index = True)
62
63      person_data.to_csv("docs/processed_data.csv")
64      person_data_young.to_csv("docs/processed_data_young.csv")
65      person_data_mid.to_csv("docs/processed_data_mid.csv")
66      person_data_old.to_csv("docs/processed_data_old.csv")
67
68  def __repr__(self):
69      return f"{self.__class__.__name__}({self.message})"
```

7.2 模型代码

Listing 3: models.py

```

# 皮尔逊相关性分析
def corr(data:pd.DataFrame, save_path, figsize = (15, 20), symmetry=True,
        x: list=None, y: list=None):
    scaler = StandardScaler()
    data_normalized = scaler.fit_transform(data)
    data = pd.DataFrame(data=data_normalized, columns=data.columns)
    data = data.corr()
    if symmetry == False:
        assert x is not None and y is not None, f'if symmetry is False, x and y
            must be given.'
        data = data.iloc[x, y]

    plt.figure(figsize=figsize)
    sns.heatmap(data, cmap='coolwarm', annot=True, linewidths=1, vmin=-1)
    plt.savefig(save_path)
    plt.clf()

# CCA 相关性分析
def cca(X, Y, save_path, figsize = (10, 15), x_label="Basic Info"):
    scaler = StandardScaler()
    X_sc = scaler.fit_transform(X) #scale data
    Y_sc = scaler.fit_transform(Y)
    cca = CCA(n_components=2)
    cca.fit(X_sc, Y_sc)
    coef_df = pd.DataFrame(np.round(cca.coef_, 2), columns = [Y.columns])
    coef_df.index = X.columns
    plt.figure(figsize=figsize)
    sns.heatmap(coef_df.T, cmap='coolwarm', annot=True, linewidths=1, vmin=-1)
    plt.xlabel(x_label)
    plt.savefig(save_path)
    plt.clf()

# XGBOOST-SHAP模型
def xgboost_shap(X: pd.DataFrame, Y:pd.DataFrame, save_path):
    # normalize
    scaler = StandardScaler()
    X_normalized = X.copy()
    X_normalized[X.columns] = scaler.fit_transform(X)

    # train the xgboot model
    X_train, X_test, y_train, y_test = train_test_split(X_normalized, Y,
        test_size=0.2, random_state=42)
```

```
    model = xgb.XGBRegressor()
    model.fit(X_train, y_train)

    # use shap to explain
    explainer = shap.TreeExplainer(model)
    shap_values = explainer.shap_values(X_test)

    # form the pics
    shap.summary_plot(shap_values, X_test, show=False)
    plt.gcf().savefig('{}_{}.png'.format(save_path, 1), bbox_inches='tight')
    plt.clf()
    shap.summary_plot(shap_values, X_test, show=False, plot_type="bar",)
    plt.gcf().savefig('{}_{}.png'.format(save_path, 2), bbox_inches='tight')
    plt.clf()

# 卡方检验
def chi_square_test(data):
    c_table = pd.crosstab(data["age_group"],data["disease"],margins=True)
    print(c_table)
    f_obs = np.array([c_table.iloc[0][0:3].values, \
        c_table.iloc[1][0:3].values,c_table.iloc[2][0:3].values])
    print(f_obs)
    print(stats.chi2_contingency(f_obs))
```

7.3 处理问题代码

Listing 4: main.py

```

from data_process import get_data, read_data
from models import histogram, ratio, cca, corr, xgboost_shap, chi_square_test
import pandas as pd
import numpy as np

# GRT DATA
def pre_work():
    read_data('data/附件2 慢性病及相关因素流调数据.xlsx')
    data = get_data()
    data.get_dataframe()

# PROBLEM-1-DRAW PICS
def problem_1():
    # ratio pic
    data = get_data()
    ratio(data.evaluate_ratio, save_path="pics/problem1/evaluate_ratio.png")
    # histogram pics
    df = pd.read_csv("docs/processed_data.csv")
    histogram(df, "qty_f_veg", [x for x in range(0, 30, 2)],
              "pics/problem1/fresh_vegtables.png", standard=[6, 30])
    histogram(df, "num_day_foods", [x for x in range(0, 20, 2)],
              "pics/problem1/num_day_foods.png", standard=[12, 20])
    histogram(df, "qty_f_fruits", [x for x in np.arange(0, 8, 0.5)],
              "pics/problem1/quantity_fresh_fruits.png", standard=[4, 7])
    histogram(df, "qty_d_prods", [x for x in np.arange(0, 8, 0.5)],
              "pics/problem1/dairy_products.png", standard=[4, 10])
    histogram(df, "qty_cereal", [x for x in np.arange(0, 5, 0.5)],
              "pics/problem1/quantity_cereal.png", standard=[1, 3])
    histogram(df, "qty_lpfem", [x for x in range(0, 15, 1)],
              "pics/problem1/quantity_lpfem.png", standard=[2.4, 4])
    histogram(df, "qty_oil", [x for x in range(0, 150, 10)],
              "pics/problem1/quantity_oil.png", standard=[25, 30])
    histogram(df, "qty_beans", [x for x in np.arange(0, 1.2, 0.05)],
              "pics/problem1/quantity_beans.png", standard=[0.6, 1])
    histogram(df, "qty_salt", [x for x in range(0, 30, 2)],
              "pics/problem1/quantity_salt.png", standard=[0, 5])
    histogram(df, "qty_wine", [x for x in np.arange(0, 50, 5)],
              "pics/problem1/quantity_wine.png", standard=[0, 15])
    histogram(df, "qty_beverage", [x for x in np.arange(0, 1.2, 0.05)],
              "pics/problem1/quantity_beverage.png", standard=[0, 1])
    histogram(df, "ratio_at_home", [x for x in np.arange(0, 1, 0.1)],
              "pics/problem1/ratio_at_home.png", standard=[0.5, 1])
```

```
44
45  # PROBLEM-2-CCA
46  def problem_2():
47      df = pd.read_csv("docs/processed_data.csv")
48      df = df.fillna(0)
49      data = df.iloc[:,np.r_[1,16:36]]
50      data.iloc[0] = False
51      data = data.replace({'True': 1, 'False': 0})
52      data = data.astype("int")
53      X1 = data.iloc[:, np.r_[0:13]]
54      X2 = data.iloc[:, np.r_[14:16]]
55      Y = data.iloc[:, np.r_[16:21]]
56      corr(data.iloc[:, np.r_[0:13, 16:21]], "pics/problem2/corr_analysis_1.png",
57          symmetry=False, x=np.r_[0:13], y=np.r_[13:18])
58      corr(data.iloc[:, np.r_[14:16, 16:21]], "pics/problem2/corr_analysis_2.png",
59          figsize=(12, 6), symmetry=False, x=np.r_[0:2], y=np.r_[2:7])
60      cca(X1, Y, "pics/problem2/CCA_1.png")
61      cca(X2, Y, "pics/problem2/CCA_2.png", figsize=(12,6))
62
63  # PROBLEM-3-XGBOOST-SHAP
64  def problem_3():
65      # read data
66      df = pd.read_csv("docs/processed_data.csv")
67      df = df.fillna(0)
68      X = df.iloc[:,np.r_[1, 16:28, 29:31]]
69      Y = df.iloc[:,np.r_[40:45, 47]]
70      # cca
71      cca(Y, X, "pics/problem3/CCA_3.png", figsize=(25, 12))
72      # corr
73      corr(Y, "pics/problem3/coor_diseases.png", figsize=(12, 12))
74      # # xgboost-shap
75      for i in range(Y.shape[1]):
76          y = Y.iloc[:,np.r_[i]]
77          xgboost_shap(X, y, "pics/problem3/"+y.columns.item())
78
79  def problem_4():
80      # chi_square_test
81      df = pd.read_csv("docs/processed_data.csv")
82      df = df.fillna(0)
83      data = df.iloc[:,-2:]
84      chi_square_test(data)
85      # read data
86      df = pd.read_csv("docs/processed_data_young.csv")
87      df = df.fillna(0)
88      X = df.iloc[:,np.r_[1, 16:28, 29:31]]
89      Y = df.iloc[:,np.r_[40:45, 46]]
90      # cca
```

```
    cca(X, Y, "pics/problem4/young/CCA_young.png", figsize=(25, 12))
    # corr
    corr(Y, "pics/problem4/young/coor_diseases_young.png", figsize=(12, 12))
    # # xgboost-shap
    for i in range(Y.shape[1]):
        y = Y.iloc[:,np.r_[i]]
        xgboost_shap(X, y, "pics/problem4/young/"+y.columns.item()+"_young")

    # read data
    df = pd.read_csv("docs/processed_data_mid.csv")
    df = df.fillna(0)
    X = df.iloc[:,np.r_[1, 16:28, 29:31]]
    Y = df.iloc[:,np.r_[40:45, 46]]
    # cca
    cca(X, Y, "pics/problem4/mid/CCA_mid.png", figsize=(25, 12))
    # corr
    corr(Y, "pics/problem4/mid/coor_diseases_mid.png", figsize=(12, 12))
    # # xgboost-shap
    for i in range(Y.shape[1]):
        y = Y.iloc[:,np.r_[i]]
        xgboost_shap(X, y, "pics/problem4/mid/"+y.columns.item()+"_mid")

    # read data
    df = pd.read_csv("docs/processed_data_old.csv")
    df = df.fillna(0)
    X = df.iloc[:,np.r_[1, 16:28, 29:31]]
    Y = df.iloc[:,np.r_[40:46, 46]]
    # cca
    cca(X, Y, "pics/problem4/old/CCA_old.png", figsize=(25, 12))
    # corr
    corr(Y, "pics/problem4/old/coor_diseases_old.png", figsize=(12, 12))
    # # xgboost-shap
    for i in range(Y.shape[1]):
        y = Y.iloc[:,np.r_[i]]
        xgboost_shap(X, y, "pics/problem4/old/"+y.columns.item()+"_old")

if __name__ == '__main__':
    pre_work()
    problem_1()
    problem_2()
    problem_3()
    problem_4()
```