



中国研究生创新实践系列大赛

中国光谷·“华为杯”第十九届中国研究生

数学建模竞赛

学 校

北京建筑大学

参赛队号

22100160013

队员姓名

1.

李佳慧

2.

邓露露

3.

郭德玉

中国光谷·“华为杯”第十九届中国研究生 数学建模竞赛

草原放牧策略研究

草原作为世界上分布最广的重要的陆地植被类型之一，在维护生物多样性、涵养水土、净化空气、固碳、调节水土流失和沙尘暴等方面具有重要的生态功能，合理的放牧政策是带动区域经济、防止草原沙漠化及保障民生的关键，放牧优化问题的研究为国家、政府制定放牧政策和草原管理决策提供科学的依据，因此，研究草原放牧策略具有重要意义。

针对问题二，建立模型对 2022 年、2023 年不同深度土壤湿度进行预测。首先对附件中的数据标准化，然后用 SPSS 进行相关性分析，计算出不同土壤湿度和影响因素之间的皮尔逊相关系数，进而选取相关性显著的因素作为不同深度土壤湿度的影响因素。其次使用土壤湿度和相应的影响因素的历史数据，利用长短期记忆（LSTM）时间序列模型对 2022 年、2023 年不同深度土壤湿度进行预测。

针对问题四，建立土壤板结化模型，给出放牧策略模型。首先对数据进行标准化，利用 R 语言 `summary` 函数对其做主成分分析，确定与沙漠化程度最具相关性的几个因素，并建立表达式。其次，选取标准化后的土壤板结化相关数据作为因素做主成分分析，分别建立不同放牧强度下土壤板结化影响因素模型。最后，将第之前预测的数据带入模型，得到不同放牧强度下土壤沙漠化、板结化程度，取其最小值即为最优放牧策略。

针对问题五，在保持草原可持续发展情况下对实验草场内放牧羊的数量进行求解，找到最大阈值。首先选取实际数据带入到问题四模型中求解参数。其次选择 0.4 和 0.3 作为沙漠化和板结化可逆的最大界限。最后使用目标规划模型，在可持续发展的前提下，利用附件 14、15 中的数据和实际数据求解最大值，即为羊群数量的最大阈值。

针对问题六，基于时间序列模型预测 2023 年 9 月份的土壤化学性质、土壤湿度和植被指数(NDVI)，并用图示的方式直观展示示范区 2012-2023 年 9 月土地状态，并分析其变化趋势。

关键词：微分方程；长短期记忆（LSTM）算法；差分整合移动平均自回归模型（ARIMA）；主成分分析；熵值法；目标规划模型

目 录

一、问题简介	5
1.1 问题背景.....	5
1.2 问题重述.....	5
二、问题分析	6
2.1 问题一分析.....	6
2.2 问题二分析.....	6
2.3 问题三分析.....	6
2.4 问题四分析.....	7
2.5 问题五分析.....	7
2.6 问题六分析.....	7
三、模型假设与符号说明.....	8
3.1 模型假设.....	8
3.2 符号说明.....	8
四、问题一模型建立与求解.....	10
4.1 数据处理.....	10
4.2 模型建立.....	11
4.3 结果与分析.....	12
4.3.1 放牧强度对土壤湿度的影响分析	12
4.3.2 放牧强度对植被生物量的影响分析	12
五、问题二模型建立与求解.....	13
5.1 数据处理.....	13
5.2 基于 LSTM 时间序列预测土壤湿度	15
5.2.1 长短期记忆 (LSTM) 模型	15
5.2.2 基于 LSTM 时间序列进行预测	17
5.3 结果与分析.....	18
六、问题三模型建立与求解.....	19
6.1 数据处理.....	19
6.2 模型建立.....	19
6.3 结果与讨论.....	20
6.3.1 放牧强度对土壤化学性质影响	20
6.3.2 2022 年土壤同期化学性质预测	23
七、问题四模型建立与求解.....	23
7.1 数据处理.....	23
7.2 模型建立.....	24
7.3 结果与讨论.....	24
7.3.1 沙漠化程度指数预测模型.....	24
7.3.2 不同放牧强度下监测点土壤板结化模型	25
7.3.3 2022 年土壤板结化程度预测	28
八、问题五模型建立与求解.....	29
8.1 数据处理.....	29
8.2 模型建立.....	29
8.2.1 假设.....	29

8.2.2 具体模型.....	30
8.3 结果与讨论.....	30
九、问题六模型建立与求解.....	31
9.1 数据处理.....	31
9.2 模型建立.....	31
9.3 结果与讨论.....	31
十、总结与评价	33
10.1 模型优点.....	33
10.2 模型缺点及改进方向.....	33
参考文献	34
附录	35

一、问题简介

1.1 问题背景

中国是世界上草原资源最丰富的国家之一，草原总面积将近 4 亿公顷，占全国土地总面积的 40%，为现有耕地面积的 3 倍。如果从中国的东北到西南划一条斜线，也就是从东北的完达山开始，越过长城，沿吕梁山，经延安，一直向西南到青藏高原的东麓为止，可以把中国分为两大地理区：东南部分是丘陵平原区，离海洋较近，气候温湿，大部分为农业区；西北部分多为高山峻岭，离海洋远，气候干旱，风沙较多，是主要的草原区。

草原作为世界上分布最广的重要的陆地植被类型之一，在维护生物多样性、涵养水土、净化空气、固碳、调节水土流失和沙尘暴等方面具有重要的生态功能。自 2003 年党中央、国务院实施“退牧还草”政策以来，在保护和改善草原生态环境、改善民生方面取得了显著成效。“退牧还草”并不是禁止放牧，除了部分区域禁牧外，很多草原实行划区轮牧以及生长季休牧。合理的放牧政策是带动区域经济、防止草原沙漠化及保障民生的关键，放牧优化问题的研究也为国家、政府制定放牧政策和草原管理决策提供科学的依据。

锡林郭勒草原位于内蒙古自治区锡林浩特市境内，面积 107.86 万公顷，1985 年经内蒙古自治区人民政府批准建立，1987 年被联合国教科文组织接纳为“国际生物圈保护区”网络成员，1997 年晋升为国家级，主要保护对象为草甸草原、典型草原、沙地疏林草原和河谷湿地生态系统。

作为我国四大草原之一，锡林郭勒草原地处内蒙古自治区，生态系统类型复杂多样、保存较为完好。当地政府为缓解草原压力、恢复生态，率先于 2015 年将基本草原、林地、水域、沙地等生态功能区、敏感区、脆弱区划入生态红线，近年来对草场实施禁牧、轮牧、休牧、严格规定载畜量等举措。至今，成效有目共睹。同时，锡林郭勒草原也是我国北方重要的绿色农畜产品输出基地，在如此生态保护政策下，畜牧业也在悄然发生变化。^[1]

内蒙古锡林郭勒草原是中国最大的草原与草甸生态系统类型的自然保护区，不仅保护了草原的生物多样性，还为发展畜牧业提供了良好的生态环境。内蒙古锡林郭勒草原是我国重要的畜牧业生产基地。放牧强度可以分为四种，分别为：对照、轻度放牧强度、中度放牧强度、重度放牧强度。适度放牧时，牲畜会采食植物的地上生物量，还会刺激植物的超补偿生长，然而，过犹不及，过度放牧会破坏植物根部，削弱了对土壤水分和营养物质的保护，导致水分流失，盐碱化加重，最终造成草场退化，土地荒漠化。因此，现代草地资源的经营应遵循可持续发展原则，在保证生态环境良性健康发展中寻求经济利益的最大化。合理的放牧政策可以防止草原沙漠化，保障牧民生活，带动区域经济。放牧优化问题的研究也为国家、政府制定放牧政策和草原管理决策提供科学的依据。

1.2 问题重述

问题 1、从机理分析的角度，建立四种放牧策略如何影响锡林郭勒草原的土壤湿度和植被生物量的数学模型

问题 2、利用土壤湿度、土壤蒸发以及降水等数据，建立模型对保持目前放牧策略不变情况下对 2022 年和 2023 年共 24 个月份的 10cm、40cm、100cm 和 200cm 的土壤湿度进行预测，并完成下表。

问题 3、从机理分析的角度，建立不同放牧策略（放牧方式和放牧强度）对锡林郭勒草原土壤化学性质影响的数学模型。并请结合附件 14 中数据预测锡林郭勒草原监测样地(12 个放牧小区)在不同放牧强度下 2022 年土壤同期有机碳、无机碳、全 N、土壤 C/N 比等值,并完成下表。

问题 4、利用沙漠化程度指数预测模型和附件提供数据（包括自己收集的数据）确定不同放牧强度下监测点的沙漠化程度指数值。并请尝试给出定量的土壤板结化定义，在建立合理的土壤板结化模型基础上结合问题 3，给出放牧策略模型，使得沙漠化程度指数与板结化程度最小。

问题 5、锡林郭勒草原请在给定的降水量（300mm，600mm、900 mm 和 1200mm）四种情形下，在保持草原可持续发展情况下对锡林郭勒草原的实验草场内放牧羊的数量进行求解，找到最大阈值。（注：这里计算结果可以不是正整数）

问题 6、在保持附件 13 的示范牧户放牧策略不变和问题 4 中得到的放牧方案两种情况下，用图示或者动态演示方式分别预测示范区 2023 年 9 月土地状态（比如土壤肥力变化、土壤湿度、植被覆盖等）。

二、问题分析

2.1 问题一分析

针对问题一，从机理分析的角度，建立不同放牧策略（放牧方式和放牧强度）对锡林郭勒草原土壤物理性质（主要是土壤湿度）和植被生物量影响的数学模型，由于缺乏放牧方式的数据，不同放牧策略主要考虑不同的放牧强度，主要为对照、轻牧、中牧和重牧。

首先搜集和整合不同放牧强度下，土壤湿度和植被生物量的变化情况，对数据进行处理。根据数据和扩展阅读给出不同放牧强度对锡林郭勒草原土壤湿度和植被生物量影响的数学模型，并结合放牧强度对不同层次土壤湿度影响的折线图和放牧强度对植被生物量影响的折线图，对不同放牧强度对土壤湿度和植被生物量的影响进行分析。

2.2 问题二分析

针对问题二，根据附件 3 土壤湿度数据、附件 4 土壤蒸发数据以及附件 8 中降水等数据，建立模型对保持目前放牧策略不变情况下对 2022 年、2023 年不同深度土壤湿度进行预测。

首先对附件中的数据进行筛选和整合，要对土壤湿度进行预测，就要先分析出不同深度土壤湿度的影响因素，因此，我们对数据进行标准化，然后用 SPSS 进行相关性分析，选取相关性显著的因素作为不同深度土壤湿度的影响因素。其次使用土壤湿度和相应的影响因素的历史数据，利用 LSTM 时间序列模型对 2022 年、2023 年不同深度土壤湿度进行预测，并对预测结果进行整合。

2.3 问题三分析

问题三一共可以分为两问：1、从机理分析的角度，建立不同放牧策略（放牧方式和放牧强度）对锡林郭勒草原土壤化学性质影响的数学模型。2、结合附件 14 中数据预测锡林郭勒草原监测样地(12 个放牧小区)在不同放牧强度下 2022 年土壤同期有机碳、无机碳、

全 N、土壤 C/N 比等值。

针对以上问题我们需要先对数据进行处理，以对应题目的后续分析。首先将数据依据时间进行排序，并将定性数据定量化处理，定量数据标准化处理。然后针对每一项化学性质单独设计一个回归模型，分别调参。并画出不同放牧强度下同一营养素的箱线图，观察不同放牧强度对同一样营养素的含量影响。

针对第二问，我们运用时间序列对不同放牧强度下各个放牧小区依据 2012-2020 年所提供的数据来对 2022 年土壤同期有机碳、无机碳、全 N、土壤 C/N 比等值进行预测。主要利用 R 语言运用 ARIMA 模型。

2.4 问题四分析

问题四可分为三个问题：1、利用沙漠化程度指数预测模型和附件提供数据（包括自己收集的数据）确定不同放牧强度下监测点的沙漠化程度指数值。2、并请尝试给出定量的土壤板结化定义。3、在建立合理的土壤板结化模型基础上结合问题 3，给出放牧策略模型，使得沙漠化程度指数与板结化程度最小。

上述问题我们需要首先对相应的数据进行处理。将与沙漠化相关的指标因子：风速、降水、气温、植被盖度、地表水资源、地下水位、人口数量、牲畜数量、社会经济水平等，根据文献[2]第五章中给出的判定标准将其标准化。土壤板结化相关因素：土壤有机物、土壤湿度和土壤的容重，运用公式将其标准化处理。

针对第一问我们将上述因素作为自变量设定为影响沙漠化程度的因素。利用 R 语言 `summary` 函数对其做主成分分析，确定与沙漠化程度最具相关性的几个因素，并建立表达式。针对第二问，我们取标准化后的土壤板结化相关数据作为因素做主成分分析，分别建立不同放牧强度下土壤板结化影响因素模型。针对第三问，将第之前预测的数据带入模型，得到不同放牧强度下土壤沙漠化、板结化程度，取其最小值即为最优放牧策略。

2.5 问题五分析

问题五可以分为两个问题：1.求草原可持续发展的程度，即土壤沙漠化和板结化可逆时的取值 2.预测在可持续发展限制下的四种降雨量的最大羊群数量。

对于第一问，可以选取实际数据带入到问题 4 的模型中求解参数实际值。本题中选择了 0.4 和 0.3 作为沙漠化和板结化可逆的最大界限。对于第二问，使用目标规划模型，在可持续发展的前提下，利用附件 14、15 中的数据和实际数据求解最大值，即为羊群数量的最大阈值。

2.6 问题六分析

问题六在保持附件 13 的示范牧户放牧策略不变和问题 4 中得到的放牧方案两种情况下，用图示或者动态演示方式分别预测示范区 2023 年 9 月土地状态（比如土壤肥力变化、土壤湿度、植被覆盖等）。

由前面问题得出轻度放牧强度是最有利于草原土壤保护的放牧策略，因此本题我们需在保持放牧策略和轻度放牧强度不变的情况下，预测 2023 年 9 月份的土壤化学性质、土壤湿度和植被指数(NDVI)。方法类似于问题三中的时间序列预测。

三、模型假设与符号说明

3.1 模型假设

考虑到现实问题，本文做出如下假设：

- (1) 假设除了附件中的变量，没有其他变量会对生物活性产生影响；
- (2) 假设附件提供的数据是真实可靠的；
- (3) 假设数据中个别缺失数据对结果不会产生重大影响。
- (4) 假设一块实验地的面积为 1 平方千米；1 平方千米草地一天能生产干草 93600Kg,其中可食用的为 40500Kg，
- (5) 假设不考虑土壤可持续发展的情况下，1 平方千米草地承载量为 260 只羊；
- (6) 假设一只羊的食量为 1.8Kg 草；
- (7) 假设土壤沙漠化程度小于 0.4，土壤板结化程度小于 0.3 时即为土壤可持续发展状态

3.2 符号说明

表 3-1 符号与符号说明

符号	符号说明	符号	符号说明
w	植被生物量	q^*	滑动平均项数
S	单位面积的载畜率	L	滞后算子
ΔW	土壤贮水变化量	SM	沙漠化程度指数
W_t	时间段内初始的土壤含水量	Q_1	平均风速强度
W_{t+1}	时间段内最后的土壤含水量	Q_2	降水量强度
P	降水量	Q_3	温度强度
G_u	地下水毛管上升量	Q_4	植被指数(NDVI)强度
G_d	土壤水渗透量	Q_5	地表水资源量强度
Et_a	实际蒸发量	η	调节系数
R_{in}	入径流量	W_{c_1}	平均风速强度权重
R_{out}	出径流量	W_{c_2}	降水量强度权重

IC_{store}	植被截流量	W_{c_3}	温度强度权重
c_p	植被覆盖率	W_{c_4}	植被指数(NDVI)强度权重
IC_{max}	特定植被的最大截流量	W_{c_5}	地表水资源量强度权重
R_{cum}	累积降雨量	W_{c_6}	人口密度强度权重
y_k	土壤湿度	B	土壤板结化程度指数
x_k^i	不同深度土壤湿度对应的影响因素	W_1	10cm 湿度(kg/m ²)
t_k^i	时间步长	W_2	40cm 湿度(kg/m ²)
W_4	200cm 湿度(kg/m ²)	W_3	100cm 湿度(kg/m ²)
a_i	所占权重	O	土壤 C/N 比
x_1	植被生物量	x_3	土壤湿度
x_2	土壤含氮量	Z	羊群数量

四、问题一模型建立与求解

4.1 数据处理

针对放牧强度对土壤物理性质的影响，主要考虑放牧强度对土壤湿度的影响。由于附件中缺乏放牧强度对土壤物理性质影响的相关数据，我们通过广泛阅读大量文献，搜集了针对华北农牧交错地带，关于放牧强度对土壤水分影响的相关数据[3]。土壤含水量一般是指土壤绝对含水量，即 100g 烘干土中含有若干克水分，与土壤湿度本质相同，只是表现方法上有所不同。具体数据见表 4-1。

表 4-1 放牧强度对土壤含水量的影响

土壤层次(cm)	放牧强度	土壤含水量(%)			
		放牧前	第 1 次牧后	第 2 次牧后	第 3 次牧后
0-10	对照	6.85	4.13	2.25	7.72
	轻牧	6.54	3.73	2.16	6.72
	中牧	6.67	4.61	2.22	8.55
	重牧	6.60	4.00	2.21	9.55
10-20	对照	7.90	6.86	3.51	8.73
	轻牧	7.94	6.09	3.28	9.25
	中牧	7.75	6.86	3.67	8.63
	重牧	8.06	6.62	3.77	7.16
20-30	对照	9.15	7.65	3.91	9.56
	轻牧	8.40	6.23	3.67	8.41
	中牧	9.01	7.64	3.68	8.01
	重牧	8.51	6.69	4.15	6.89

针对放牧强度对植被生物量的影响，根据内蒙古锡林郭勒草原数据及其说明，采用收割法齐地面收割地上植物部分，收集到的植物同天晚上称取鲜重后，置于 65℃的鼓风烘箱中，24 小时后，称取干重，即为植物地上生物量，因此，我们整合附件 15 中放牧强度和植被干重的数据，针对不同放牧强度，对牧前、第一次牧后、第二次牧后、第三次牧后以及第四次牧后的各种植被干重进行求和，作为植被生物量，整合数据见表 4-2。

表 4-2 放牧强度对植被生物量的影响

放牧强度	植被生物量				
	放牧前	第 1 次牧后	第 2 次牧后	第 3 次牧后	第 4 次牧后
对照	220.22	1473.68	2053.52	1749.62	1906.7
轻牧	318.91	983.02	1197.74	1282.05	1437.67
中牧	321.86	1011.45	1142.56	1241.89	1099.95
重牧	265.49	659.12	987.72	726.86	485.13

4.2 模型建立

根据扩展阅读我们可以得到，对于放牧与植物生长之间的关系，Woodward 等建立了如下一个简单模型^[4]：

$$\frac{dw}{dt} = 0.049w(1 - \frac{w}{4000}) - 0.0047Sw$$

式中， w 为植被生物量， S 为单位面积的载畜率。

该模型只考虑放牧影响，简单地反映了载畜率对植被生物量的作用，没有考虑其他因素的影响，只是从某一个侧面刻某一个因素对于植被生长的影响。

草原水分通过降水、入渗、蒸发等形式周而复始的循环，其过程十分复杂，在无人干扰的情况下，土壤-植被-大气系统的水平衡基本方程为

$$\Delta W = W_{t+1} - W_t = P + G_u + R_{in} - (Et_a + G_d + R_{out} + IC_{store})$$

其中， ΔW 为土壤贮水变化量， W_{t+1} 和 W_t 分别为时间段内的始末土壤含水量， P 为降水量， G_u 和 G_d 分别为地下水毛管上升量和土壤水渗透量， Et_a 为实际蒸发量， R_{in} 和 R_{out} 分别为入和出径流量， IC_{store} 为植被截流量。锡林郭勒草原地势比较平坦，降水量和降水强度较少，水分循环以垂直方向的水量交换为主，绝大部分降水被蓄积在土壤中，尽管在遇到较大降水时会产生局地径流，但仍在整个草原区域内，其出入径流可视为相等。

草地的植被直接决定放牧的强度，而植被的截流量能最好反映植被的生长能力，依照递推关系，放牧强度与植被的截流量存在正相关关系。

植被截流量与降水量、植被覆盖度、叶面积指数（LAI）等密切相关。植被覆盖率是植物群落覆盖地表状况的一个综合量化指标，能够直观反映地表植被的丰度。降水量小、植被覆盖度高、LAI 大时植被截流量大，其表达式为：

$$IC_{store} = c_p \cdot IC_{max} \cdot \left[1 - \exp\left(\frac{-k \cdot R_{cum}}{IC_{max}}\right) \right]$$

上式中， IC_{store} 为植被截流量（mm）； c_p 为植被覆盖率； IC_{max} 特定植被的最大截流量（mm）； k 为植被密度校正因子，与 LAI 有关； R_{cum} 为累积降雨量（mm）。 IC_{max} 可以通过 LAI 来估算：

$$IC_{max} = 0.935 + 0.498 \cdot LAI - 0.00575 \cdot LAI^2$$

式中，LAI 为一个分布式的时变参数。

根据以上知识，我们可以建立不同放牧强度对锡林郭勒草原土壤物理性质（主要是土壤湿度）和植被生物量影响的数学模型：

$$\begin{cases} \frac{dw}{dt} = aw(1 - \frac{w}{b}) - cS(t)w(t) \\ \Delta W = P(t) - Et_a(t) - IC_{store}(t) + Gu - Gd + R_{in} - R_{out} \\ IC_{store}(t) = c_p \cdot IC_{max}(t) \left[1 - \exp\left(\frac{-k \cdot R_{cum}}{IC_{max}}\right) \right] \\ IC_{max}(t) = 0.935 + 0.498 \cdot LAI(t) - 0.00575 \cdot LAI(t)^2 \end{cases}$$

4.3 结果与分析

4.3.1 放牧强度对土壤湿度的影响分析

根据 4.1 节搜集和整合的数据，我们可以得到在不同放牧强度（对照、轻牧、中牧、重牧）下，不同层次土壤湿度在牧前和三次牧后的变化情况，从而直观地分析放牧强度对不同层次土壤湿度的影响。

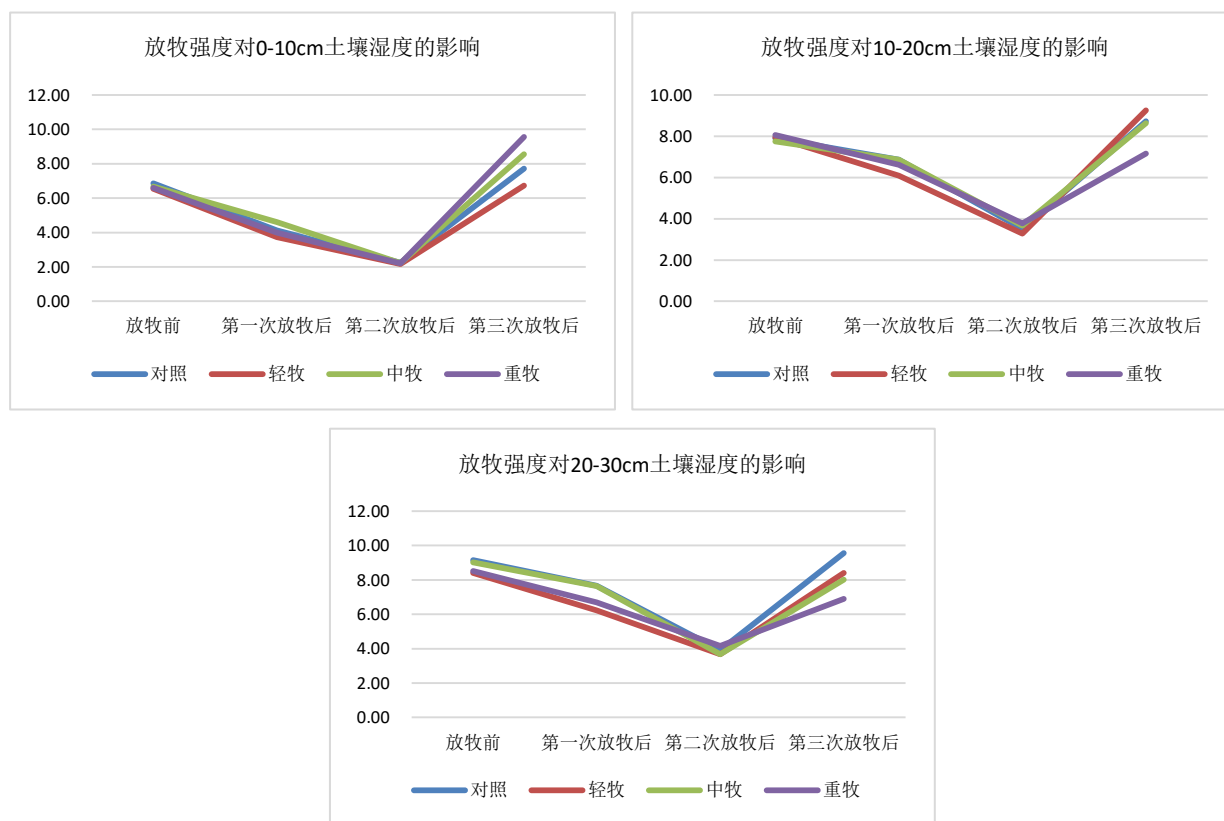


图 4-1 放牧强度对不同层次土壤湿度影响的折线图

根据分析表 4-1 和图 4-1 中数据我们可以得到以下结论：

（1）对比牧前和牧后不同层次土壤的含水量，第 1 次放牧后和第 2 次放牧后的结果表明，不同的放牧强度都使得土壤湿度呈现下降的趋势，但在第 3 次放牧后出现了相反的趋势，土壤湿度增加了，甚至超过了牧前的土壤湿度。这说明不同放牧强度在第 1 次和第 2 次放牧后都会降低土壤湿度，但继续放牧，反而使得土壤湿度增加。

（2）对比不同层次土壤湿度的变化趋势发现 0-10cm 土壤湿度对放牧强度较为敏感，在牧前，放牧强度为对照的土壤含水量最高，放牧强度为轻牧和中牧的土壤含水量次之，放牧强度为重牧的土壤含水量最低；但在第 3 次牧后，对照组的土壤湿度反而最低，重牧区的土壤湿度最高。而 10-20cm 和 20-30cm 的层次的土壤湿度在重牧区是最低的，对照组和轻牧区的土壤湿度相对较高，说明在较深的土壤区域，不同程度的放牧强度对土壤湿度的影响趋势大体上一致。

4.3.2 放牧强度对植被生物量的影响分析

根据 4.1 节搜集和整合的数据，我们可以得到在不同放牧强度（对照、轻牧、中牧、重牧）下，植被生物量在牧前和三次牧后的变化情况，从而直观地分析放牧强度对植被生物量的影响。

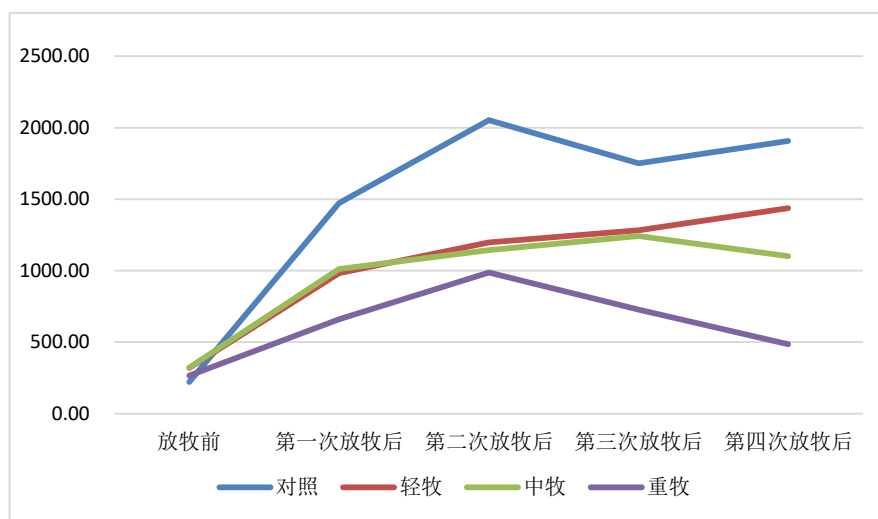


图 4-2 放牧强度对植被生物量影响的折线图

根据分析表 4-2 和图 4-2 中数据我们可以得到以下结论：

（1）对比不同放牧强度下植被生物量，在放牧前，四种放牧强度区域的植被生物量差不多，在放牧后，对照区的植被生物量始终最高，轻牧和中牧区的植被生物量居中，重牧区的植被生物量始终最少，这表明随着放牧强度的增加会在一定程度上阻碍植被生物量的增长。

（2）对比四次放牧前后的植被生物量，第 1 次和第 2 次放牧后的植被生物量对比放牧前都有所增长，说明放牧在一定程度上会刺激植物的生长，导致植物生物量增多，但在第 3 次和第 4 次牧后，植被生物量都有一定程度的下降趋势，说明过度放牧会在一定程度上一直植物的生长。

五、问题二模型建立与求解

5.1 数据处理

根据附件 3 土壤湿度数据，我们可以分别得到 2021 年 1 月-2022 年 3 月 10cm、40cm、100cm 和 200cm 这四个深度处土壤湿度(kg/m^2)随着时间的变化；根据附件 4 土壤蒸发数据我们可以得到 2021 年 1 月-2022 年 3 月土壤蒸发量(mm)随着时间的变化；根据附件 8 我们可以得到 2021 年 1 月-2022 年 3 月气温、降水、气压、能见度、风速等气候环境因素随着时间的变化。

在放牧策略不变情况下，土壤蒸发量、气温、降水、气压、能见度、风速等都会对不同深度的土壤湿度产生不同程度的影响。因此，为了确定每个深度土壤湿度的影响因素，我们先对所有数据依据时间节点进行整合，接着对数据进行标准化，根据标准化后的数据利用 SPSS 计算皮尔逊相关系数进行相关性分析，图 5-1、5-2、5-3、5-4 分别展示了土壤

蒸发量、气温、降水、气压、能见度、风速等因素对 10cm、40cm、100cm 和 200cm 深度土壤湿度的影响程度，我们根据得到的相关性系数筛选出每个深度土壤湿度的影响因素，进而依据土壤湿度和相应影响因素的历史数据对 2022 年、2023 年不同深度土壤湿度进行预测。

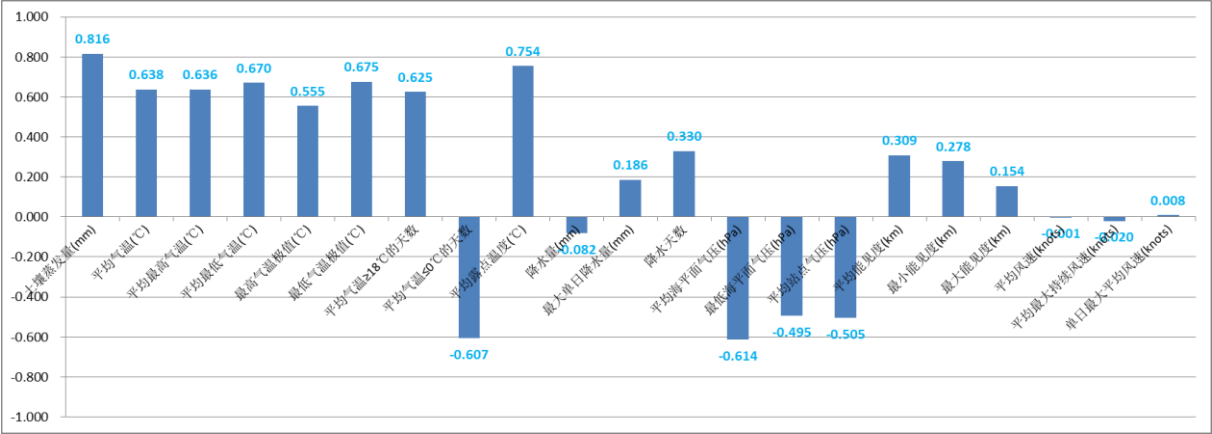


图 5-1 10cm 土壤湿度的影响因素的相关性系数

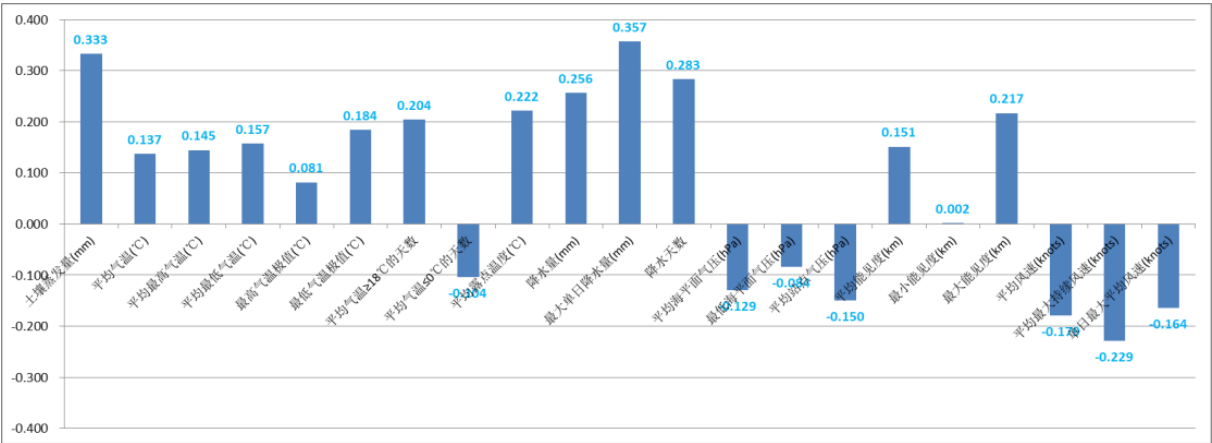


图 5-2 40cm 土壤湿度的影响因素的相关性系数

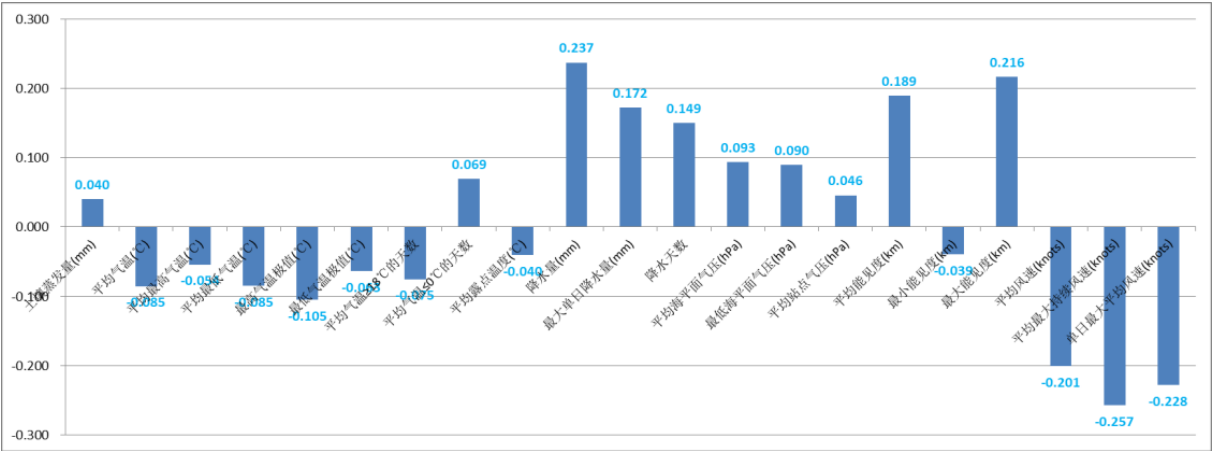


图 5-3 100cm 土壤湿度的影响因素的相关性系数

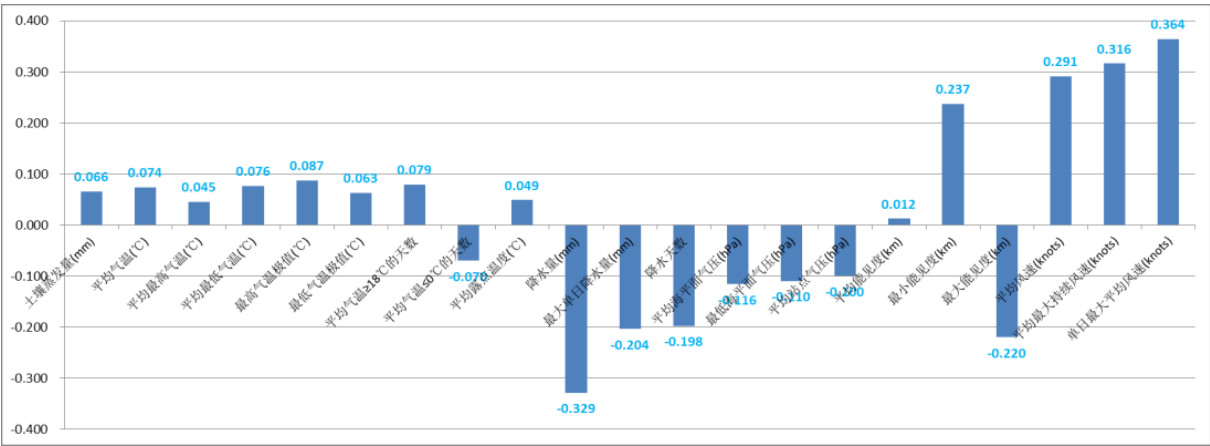


图 5-4 200cm 土壤湿度的影响因素的相关性系数

根据各个影响因素和不同土壤湿度的相关性分析得出，四个深度土壤湿度相应的影响因素如下表所示：

表 5-1 不同深度土壤湿度的影响因素

不同层次 土壤湿度	影响因素
10cm 土壤湿度 (kg/m2)	土壤蒸发量(mm)、平均气温(°C)、平均最高气温(°C)、平均最低气温(°C)、最高气温极值(°C)、最低气温极值(°C)、平均气温≥18°C的天数、平均气温≤0°C的天数、平均露点温度(°C)、降水天数、平均海平面气压(hPa)、最低海平面气压(hPa)、平均站点气压(hPa)、平均能见度(km)、最小能见度(km)
40cm 土壤湿度 (kg/m2)	土壤蒸发量(mm)、降水量(mm)、最大单日降水量(mm)、降水天数
100cm 土壤湿度(kg/m2)	降水量(mm)、平均最大持续风速(knots)
200cm 土壤湿度(kg/m2)	降水量(mm)、最小能见度(km)、平均风速(knots)、平均最大持续风速(knots)、单日最大平均风速(knots)

5.2 基于 LSTM 时间序列预测土壤湿度

5.2.1 长短期记忆（LSTM）模型

与传统的深度神经网络相比,RNN的一个改进是它将隐藏层视为连续的循环层(图 5-5, 左)。这种结构可以展开, 以产生与输入时间序列相对应的离散时间步长处神经元分量序列的输出(图 5-5, 右图)。通过这种架构, RNN 具有很强的能力, 可以捕获嵌入在输入在过去元素中的历史信息一段时间^[5]。

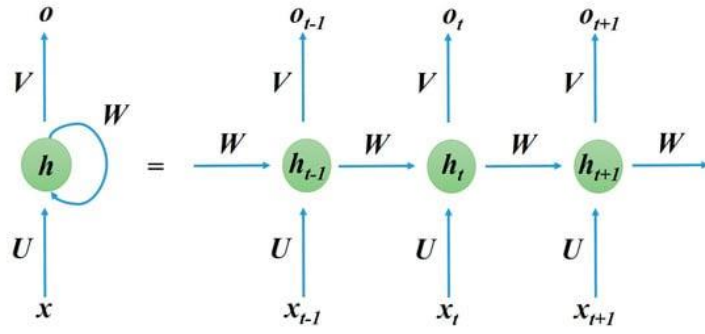


图 5-5 递归神经网络及其展开结构

图 5-5 显示了一个 RNN 的示意图。向前传递记忆的过程可以用数学方法来描述：

$$h_t = \sigma(Wx_t + Uh_{t-1} + b^{(h)})$$

$$o_t = \sigma(Vh_t + b^{(o)})$$

其中 x_t 是一个输入向量， h_{t-1} 是 RNN 中之前的隐藏状态， h_t 是对应于 x_t 的隐藏状态， o_t 是输出向量， W ， U ， V 是权值矩阵， b 表示偏置向量， σ 是一个激活函数。

RNN 的架构使得它能够基于当前时间步长的输入数据对时间序列中的下一个时间步长进行预测。然而，利用长时间序列的反向传播来训练 RNN 却具有挑战性。反向传播梯度在每个时间步长的生长或收缩可以累积。这种积累会导致梯度在许多时间步长^[5]中爆炸或消失。引入长短期记忆来克服这一挑战，以确保 RNN 具有长期记忆能力^[6]。长短期记忆允许网络使用一个特殊的隐藏单元，称为 LSTM 单元，而不是一个简单的 RNN 单元(图 5-5 中的绿色节点 h)长时间获取信息。

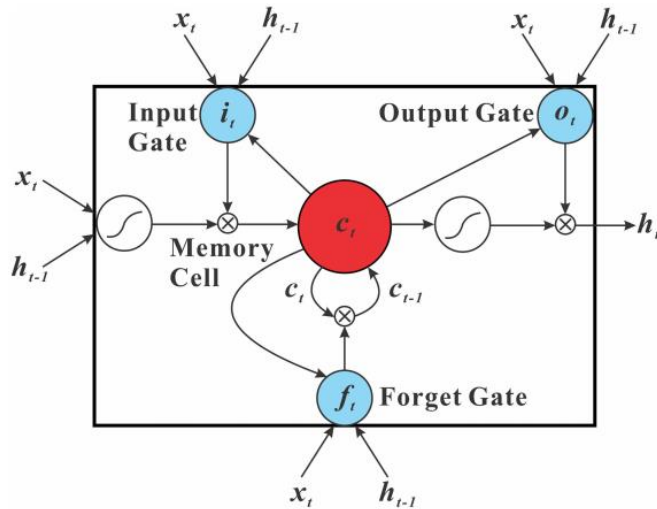


图 5-6 一个 LSTM 单元的结构

LSTM 是 RNN 的一种特殊变体。通过维持一个内部记忆细胞，LSTM 可以在长期和短期内捕获一个序列内的相关性。一个 LSTM 单元的结构如图 5-6 所示。记忆状态 c_t 位于结

构的中心,用红色圆圈表示。在一个 LSTM 中有三个门,即输入门 i_t 、遗忘门 f_t 和输出门 o_t 。它们在图中用蓝色的圆圈表示。LSTM 的数学模型:

$$\begin{aligned} i_t &= \sigma(W_{hi}h_{t-1} + W_{xi}X_t + b_i) \\ f_t &= \sigma(W_{hf}h_{t-1} + W_{xf}X_t + b_f) \\ o_t &= \sigma(W_{ho}h_{t-1} + W_{xo}X_t + b_o) \\ \tilde{c}_t &= \tanh(W_{hc}h_{t-1} + W_{xc}X_t + b_c) \\ c_t &= f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t \\ h_t &= o_t \otimes \tanh(c_t) \end{aligned}$$

其中 X_t 是第 t 个时间步的输入向量, \tilde{c}_t 候选记忆状态, σ 是 *sigmoid* 型激活函数, \tanh 表示双曲切线激活函数, W_{hi} , W_{xi} , W_{hf} , W_{xf} , W_{ho} , W_{xo} , W_{hc} , W_{xc} 为权重矩阵, b_i , b_f , b_o , b_c 为偏差项, \otimes 代表两个向量的逐点乘法。

具体地说,输入向量由输入门和遗忘门进行处理,以确定如何更新旧的内存状态 c_{t-1} 。

然后,通过使用输出门过滤标准化的 c_t , 发出输出状态(也称为“隐藏状态”) h_t 。LSTM 网络是由 LSTM 细胞组成的递归神经网络,能够学习长期时间序列中的波动模式,然后做出预测。

5.2.2 基于 LSTM 时间序列进行预测

根据 5.1 节处理之后的数据,对于不同深度的土壤湿度 y_k (k 表示 10cm、40cm、100cm 和 200cm 四个深度), 有影响因素 $x_k^1, x_k^2, \dots, x_k^m$ (m 为不同深度土壤湿度对应的影响因素的个数), LSTM 模型的输入为 123 个时间步长(从 2012 年 1 月到 2022 年 3 月) $m+1$ 维数据, 输出为 21 个时间步长(从 2022 年 4 月到 2023 年 12 月)。

1. 训练

鉴于滑动时间窗口的方法^[7], 从 $m+1$ 维 123 个时间步长的历史数据中获取用于训练 LSTM 模型的多个样本。针对土壤湿度 y_k , 时间序列 $(t_k^1, t_k^2, \dots, t_k^n)$ 具有 n ($n=123$) 个时间步长, 其中 $t_k^i = [y_k^i, x_k^{i1}, x_k^{i2}, \dots, x_k^{im}]$ 。假设时间窗口的大小为 l ($l < n$), 则第一个样本 $(t_k^1, t_k^2, \dots, t_k^l) = ([y_k^1, x_k^{11}, x_k^{12}, \dots, x_k^{1m}], [y_k^2, x_k^{21}, x_k^{22}, \dots, x_k^{2m}], \dots, [y_k^l, x_k^{l1}, x_k^{l2}, \dots, x_k^{lm}])$ 是输入子序列

用于输出 $(y_k^{l+1}, y_k^{l+2}, \dots, y_k^{l+21})$ 。时间窗口一次向前滑动一个时间步长以生成一个训练样本，因此，可以获得 $n-l-21$ 个样本来训练 LSTM 网络。此外，均方误差（MSE）损失函数和 Adam 优化器用于 LSTM 训练。

2. 预测

根据上一节中描述的训练方法，用 $n-l-21$ 个子样本训练得到一个 LSTM 网络，用该网络结合历史数据进行预测，可以得到 21 个时间步长（从 2022 年 4 月至 2023 年 12 月）的预测结果。

5.3 结果与分析

根据上述模型进行预测得到结果如下表所示：

表 5-2 2022 年、2023 年不同深度土壤湿度的预测结果

年份	月份	10cm 湿度 (kg/m ²)	40cm 湿度 (kg/m ²)	100cm 湿度 (kg/m ²)	200cm 湿度 (kg/m ²)
2022	04	13.920	41.671	52.369	166.472
	05	15.019	36.763	50.223	166.409
	06	15.694	35.313	50.000	166.371
	07	16.082	36.078	50.883	166.258
	08	16.486	36.686	52.500	166.270
	09	16.190	37.135	54.325	166.345
	10	15.692	37.427	56.623	166.336
	11	14.926	39.863	57.836	166.261
2023	12	14.691	41.132	59.654	166.148
	01	14.800	43.042	61.595	166.062
	02	14.807	44.630	63.359	165.996
	03	15.019	46.005	65.246	165.937
	04	15.888	48.406	67.209	165.786
	05	16.841	49.159	68.850	165.593
	06	17.282	51.579	70.604	165.437
	07	17.413	53.670	72.244	165.258
	08	17.713	58.405	73.894	165.190
	09	17.958	58.797	75.679	165.017
	10	17.302	59.175	77.256	164.940
	11	16.510	61.207	79.019	164.842
	12	15.942	60.200	80.568	164.736

六、问题三模型建立与求解

6.1 数据处理

针对以上问题我们需要先对附件 14 中内蒙古自治区锡林郭勒盟典型草原不同放牧强度土壤碳氮监测数据集（2012 年 8 月 15 日-2020 年 8 月 15 日）进行处理，以对应题目的后续分析。首先将数据在四种放牧强度（对照 NG、轻度放牧强度 LGI、中度放牧强度 MGI 和重度放牧强度 HGI）下对 12 个放牧小区（G17、G19、G21、G6、G12、G18、G8、G11、G16、G9、G13、G20）的 5 种土壤化学性质（土壤有机碳 SOC、土壤无机碳 SIC、土壤全碳 STC、全 N、土壤 C/N 比）依据时间进行排序。

放牧强度数据化表示：对照 NG 为 0，轻度放牧强度 LGI 为 2、中度放牧强度 MGI 为 4 和重度放牧强度 HGI 为 8，并对其标准化结果为：对照 NG 为 0，轻度放牧强度 LGI 为 0.25、中度放牧强度 MGI 为 0.5 和重度放牧强度 HGI 为 1。对应土壤化学性质，运用公式(元素含量-元素含量最小值)/(元素含量最大值-元素含量最小值)，将其标准化处理。

6.2 模型建立

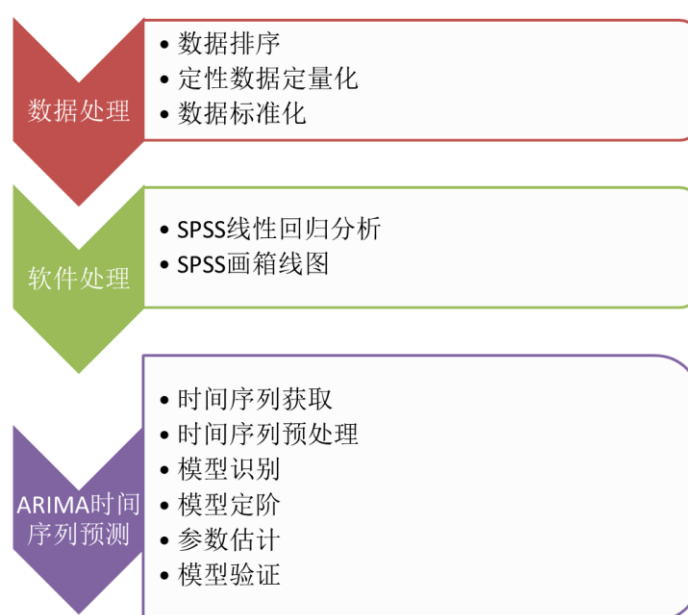


图 6-1 问题三建模流程图

对于第一个问题，需要建立不同放牧策略（放牧方式和放牧强度）对锡林郭勒草原土壤化学性质影响。数据划定标准中放牧方式固定为选择划区轮牧，则只需要考虑放牧强度对土壤化学性质影响，因此我们将放牧强度定为自变量，将土壤化学性质定为因变量，来建立线性化模型。放牧强度定这一自变量在附件 14 中为定性数据，根据题目中给定的选择划区轮牧的放牧强度标准规定：对照（NG，0 羊/天/公顷）、轻度放牧强度（LGI，2 羊/天/公顷）、中度放牧强度（MGI，4 羊/天/公顷）和重度放牧强度（HGI，8 羊/天/公顷）。将放牧强度定量化表示：对照 NG 为 0，轻度放牧强度 LGI 为 2、中度放牧强度 MGI 为 4 和重度放牧强度 HGI 为 8，并对其标准化结果为：对照 NG 为 0，轻度放牧强度

LGI 为 0.25、中度放牧强度 MGI 为 0.5 和重度放牧强度 HGI 为 1。土壤化学性质这一因变量在附件 14 中体现为定量数据，因此只需将五组数据标准化并根据时间与自变量对应。对于上述数据建立五组回归模型，分别调参得到相应表达式结果。在保持五组自变量和因变量不变的情况下画出不同放牧强度下各类土壤化学元素的箱线图，每类放牧强度下观察百分之五十分位数的变动情况以体现放牧强度对各类土壤化学元素的影响。

对于第二个问题，利用 R 语言运用差分整合移动平均自回归模型^[8]—ARIMA 模型：：1、数据可视化，识别平稳性。2、对非平稳的时间序列数据，做差分，得到平稳序列。3、建立合适的模型。平稳化处理后，若偏自相关函数是截尾的，而自相关函数是拖尾的，则建立 AR 模型；若偏自相关函数是拖尾的，而自相关函数是截尾的，则建立 MA 模型；若偏自相关函数和自相关函数均是拖尾的，则序列适合 ARMA 模型。4、模型的阶数在确定之后，对 ARMA 模型进行参数估计，比较常用是最小二乘法进行参数估计。5、假设检验，判断（诊断）残差序列是否为白噪声序列。6、利用已通过检验的模型进行预测。时间序列数据由上述过程处理得到，安装 forecast 程序包，调用 auto.arima() 函数进行系统自动定阶，确定自回归项数 p 、使之成为平稳序列所做的差分次数（阶数） d 和滑动平均项数 q 。带入 ARIMA(p, d, q) 模型：

$$\left(1 - \sum_{i=1}^p \phi_i L^i\right) (1-L)^d X_t = \left(1 - \sum_{i=1}^q \theta_i L^i\right) \varepsilon_t$$

其中 L 是滞后算子， $d \in \mathbb{Z}, d > 0$ 。

6.3 结果与讨论

6.3.1 放牧强度对土壤化学性质影响

放牧强度对土壤有机碳 SOC 模型： $y = -0.052x + 0.485$

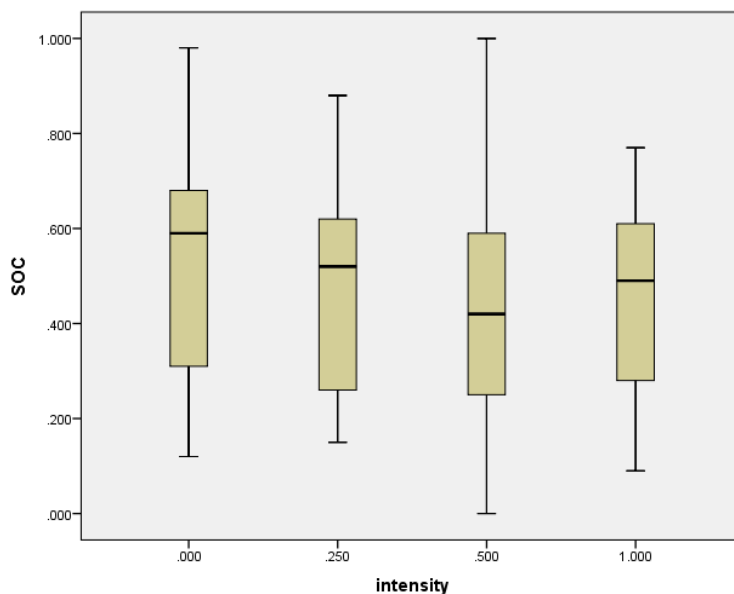


图 6-2 不同放牧强度下土壤有机碳 SOC 含量箱线图

放牧强度对土壤无机碳 SIC 模型： $y = -0.015x + 0.495$

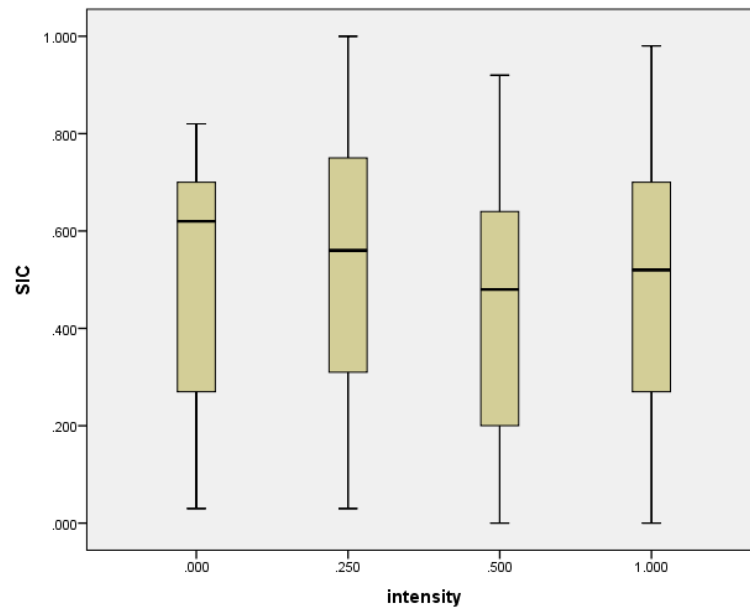


图 6-3 不同放牧强度下土壤无机碳 SIC 含量箱线图

放牧强度对土壤全碳 STC 模型： $y = -0.064x + 0.461$

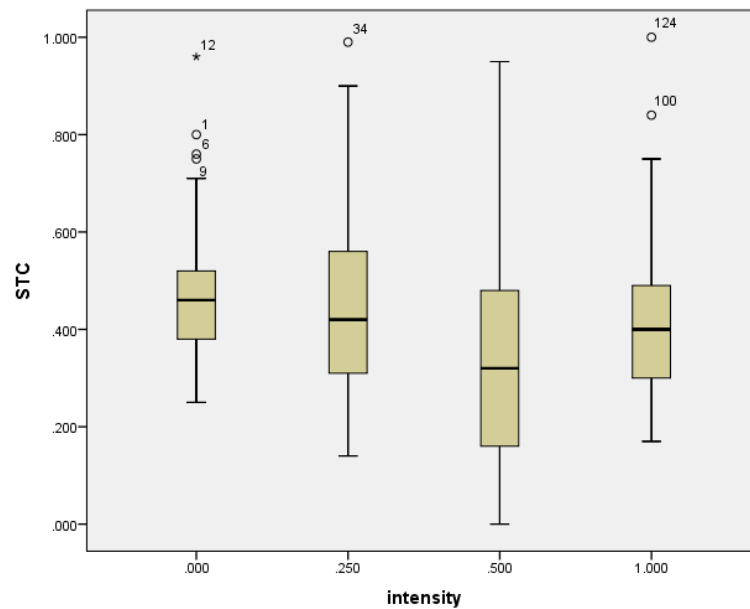


图 6-4 不同放牧强度下土壤全碳 STC 含量箱线图

放牧强度对土壤全 N 模型： $y = -0.021x + 0.509$

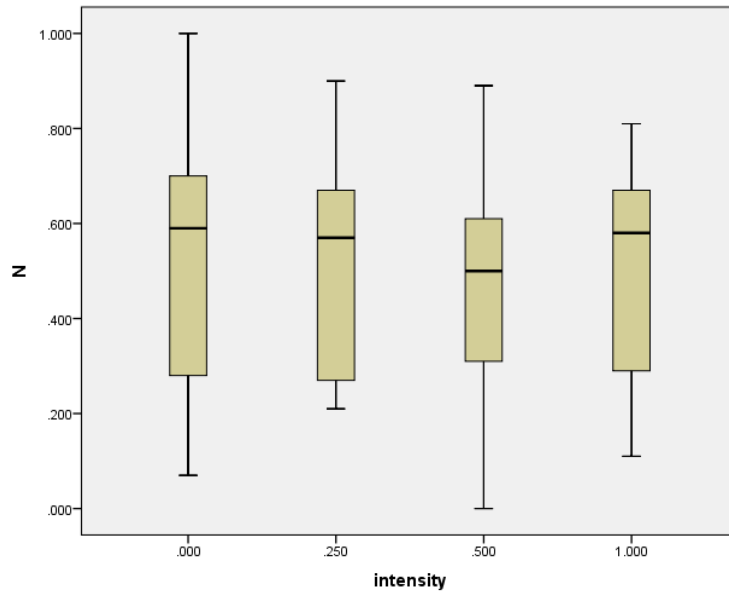


图 6-5 不同放牧强度下土壤全 N 含量箱线图

放牧强度对土壤 C/N 比模型： $y = -0.027x + 0.387$

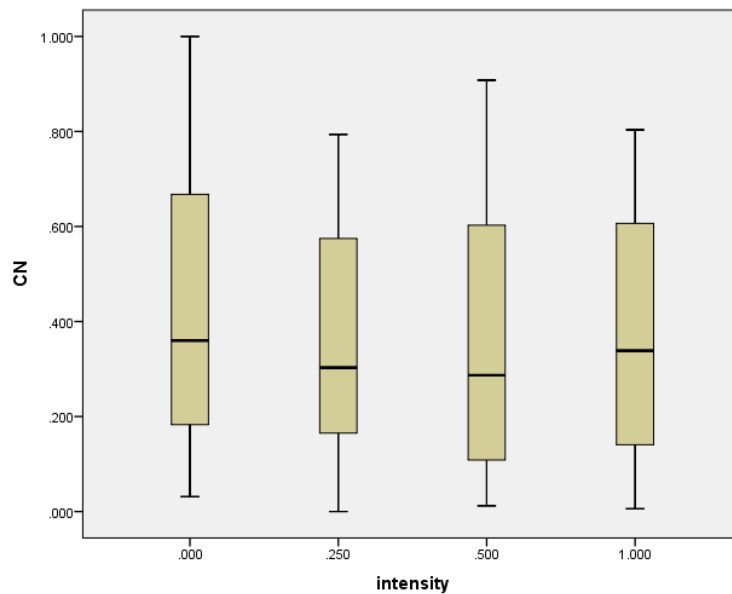


图 6-6 不同放牧强度下土壤 C/N 比箱线图

由于动物的选择性采食、踩踏和植被自身恢复能力的差异以及家畜排泄物的影响，不同放牧强度改变了植物群落的盖度、物种组成、植物功能群组成、生物量、根冠比、凋落物数量、根际沉积、土壤微生物、土壤酶活性、土壤自养呼吸等生物因素与土壤温度、土壤含水量、土壤 pH 值、土壤容重、土壤孔隙度、土壤团聚体结构、土壤异养呼吸等非生物因素，而上述因子影响着土壤化学性质的输入与分配、转化与利用以及输出，土壤有机碳在多因子的共同作用下保持动态平衡或发生着变化^[9]。根据所建模型观察得土壤化学

物质含量随放牧强度的增加而逐渐减小，但会出现波动，这是由于土壤化学性质不是由某一因素决定的，会与其他因素共同作用。这些性质深刻影响土壤的形成与发育过程，对土壤的保肥能力、缓冲能力、自净能力和养分循环等也有显著影响^[12]。

6.3.2 2022 年土壤同期化学性质预测

运用 ARIMA 模型对每类放牧强度对应的放牧小区在 2012-2020 年的时间序列基础上预测 2022 年土壤同期有机碳、无机碳、全 N、土壤 C/N 比等值。

表 6-1 不同放牧强度下 2022 年土壤同期有机碳、无机碳、全 N、土壤 C/N 比等值

放牧强度	放牧小区	土壤有机碳 SOC	土壤无机碳 SIC	土壤全碳 STC	全 N	土壤 C/N 比
NG	G17	14.030	8.876	24.024	1.825	14.127
	G19	15.800	7.101	22.900	1.811	13.562
	G21	15.815	7.055	22.871	1.902	12.978
LGI	G6	14.091	7.736	17.790	1.690	12.670
	G12	14.645	7.215	21.861	1.796	12.685
	G18	15.615	9.004	24.619	1.879	13.744
MGI	G8	14.324	5.802	16.330	1.712	11.525
	G11	14.405	5.630	20.036	1.705	11.474
	G16	13.675	9.815	23.491	1.609	15.590
HGI	G9	14.607	3.120	20.254	1.794	9.080
	G13	14.533	6.896	21.430	1.775	12.763
	G20	15.075	8.408	23.483	2.037	13.205

例如对应 NG 放牧强度下 G17 放牧小区得土壤有机碳含量进行预测：

对预处理的时间序列自动定阶，确定其自回归项数为 1、使之成为平稳序列所做的差分次数（阶数）为 0、滑动平均项数为 0。带入 ARIMA 模型当中运行得出其 forecast 值为 14.03021，即为 2022 年 NG 强度下 G17 小区的土壤有机碳 SOC 含量。

七、问题四模型建立与求解

7.1 数据处理

上述问题我们需要首先对相应的数据进行处理。由沙漠化程度指数预测模型表达式，我们可以得到沙漠化相关关系的指标因子可以考虑三个气象因素：风速、降水、气温（附件 8 数据）；三个地表因素：植被盖度（附件 6 数据）、地表水资源、地下水位（自查数据）；三个人文因素：人口数量、牲畜数量、社会经济水平（自查数据）等。由于数据缺失我们只能得到五个因素相关数据，即风速、降水、气温、植被盖度、地表水资源、人口密度。并根据文献[2]第五章中给出的判定标准将其强度数据套用条件格式判别其取值。例如对于降水因子，由于降水量与沙漠化程度成成反比，所以降水量小于 10mm 时，强度取值 1，大于 100mm 时，强度取值为 0；介于两者之间的强度取值为(100mm-降水

量)/(100mm-10mm)^[12]。由于数据缺少，取影响土壤板结化的因素我们仅仅取土壤 C/N 比及土壤各深度下湿度，我们将土壤湿度与土壤化学性质之间根据时间串联，计算 2012-2020 年不同土壤深度下年平均土壤湿度，4 种放牧强度下同时不同地的土壤 C/N 比化学性质的平均值。并运用公式将其标准化处理用于后续使用。

7.2 模型建立



图 7-1 问题四建模流程图

对于上述处理好的数据主要运用 R 语言进行主成分分析。主成分分析是一种通过降维技术把多个变量化为少数几个主成分的统计变量分析方法。研究如何通过少数几个主成分来揭示多个变量间的内部结构，即从原始变量中导出少数几个主成分，使它们尽可能多地保留原始变量的信息，且彼此间互不相关。其主要流程：1、指标数据标准化；2、指标之间的相关性判定； 3. 确定主成分个数 m ； 4. 主成分表达式； 5. 主成分命名^[12]。

对于第一问，将原来 6 个指标作线性组合，作为综合指标。用 $comp.1$ 表示选取的第一个线性组合，即第一个综合指标，其累计贡献率越大，表示 $comp.1$ 包含的信息越多。如果第一主成分不足以代表原来 6 个指标的信息，再考虑选取 $comp.2$ 即选第二个线性组合，为了有效地反映原来信息， $comp.1$ 已有的信息就不需要再出现在 $comp.2$ 中，直至累计贡献率达 90% 以上，即可将线性组合定为主成分。对于第二问，将串联起的 5 个变量，重复第一问流程得到累计贡献率达 90% 以上的主成分，分别对应不同放牧强度下的土壤板结化模型。

最后，只需将之前预测数据结果作为主成分值带入式子中去，通过比较即可得到放牧策略模型，使得沙漠化程度指数与板结化程度最小。

7.3 结果与讨论

7.3.1 沙漠化程度指数预测模型

由文献[5]可得，模型的调节系数 $\eta=1.0193x-0.0198$ ，其中 x 表示预测年度的沙漠化程度指数，其取值范围为 0-1，划分标准依据下表 7-1 给出。

表 7-1 沙漠化程度及沙漠化程度指数划分标准

划分内容		划分类型			
沙漠化程度	非沙漠化	轻度沙漠化	中度沙漠化	重度沙漠化	极重度沙漠化
沙漠化程度指数	[0,0.20]	(0.20,0.40]	(0.40,0.60]	(0.60,0.80]	(0.80,1.00]

通过主成分分析，从 6 个因子中选取主成分，得到 comp.1，带入沙漠化程度指数预测模型表达式：

$$SM = \eta \cdot \sum_{i=1}^n S_{Q_i} = \eta \cdot \sum_{i=1}^n (Q_i \cdot W_{c_i})$$

其中， SM 沙漠化程度指数， $Q_1—Q_6$ 分别表示降水量、温度、植被指数(NDVI)、地表水资源量、人口密度强度， $W_{c_1}—W_{c_6}$ 分别表示以上强度权重。

可得：

$$SM = \eta(0.563Q_1 + 0.801Q_2 - 0.206Q_4)$$

其累积贡献率为 62.764%，达不到预期结果。则继续运行得到 comp.2，其表达式为：

$$SM = \eta(0.764Q_1 - 0.599Q_2 - 0.241Q_4)$$

其累积贡献率为 96.128%，可得预期结果，因此影响沙漠化程度指数的主成分为风速、降水、植被盖度。若仍继续筛查可得以下 comp.3 结果：

$$SM = \eta(0.316Q_1 + 0.946Q_4)$$

其累积贡献率达到 99.448%，最具相关性的因素为风速和植被盖度。若进一步筛查，可得 comp.4 结果：

$$SM = \eta(0.976Q_5 + 0.208Q_6)$$

其累积贡献率为 99.784%，与 comp.3 差异不大，因此到 comp.3 即可结束筛查分析。其所有线性组合结果累计贡献率如下图 7-2 所示：

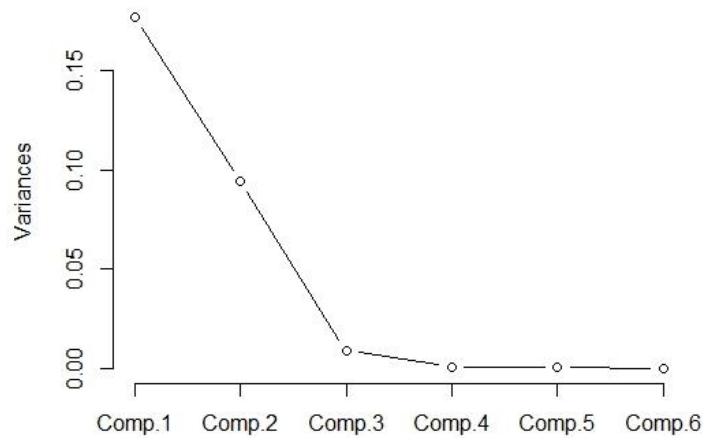


图 7-2 沙漠化程度指数所有线性组合累计贡献率折线图

由上图更加清晰得到筛查到 comp.2，其结果已经精确到 90% 以上，后续操作可简化省略。

7.3.2 不同放牧强度下监测点土壤板结化模型

土壤板结化与土壤有机物、土壤湿度和土壤的容重有关，目前还没有明确的定量表达式，其数学模型可定性描述为如下：

$$B = f(W, C, O)$$

土壤湿度 w 越少，容重 c 越大，有机物含量 o 越低，土壤板结化程度 B 越严重。其中，在建立的具体模型表达式中 B 表示土壤板结化程度指数， $W_1—W_4$ 分别表示不同层次土壤湿度， O 表示土壤 C/N 比。

对于对照组 NG 放牧强度，通过主成分分析，从 5 个因子中选取主成分，得到 comp.1，带入土壤板结化表达式可得：

$$B = 0.165W_2 + 0.166W_3 - 0.421W_4 - 0.874O$$

其累积贡献率为 94.424%，可得预期结果，因此在对照组影响土壤板结化的主成分为 40cm 湿度、100cm 湿度、200cm 湿度和土壤 C/N 比。

若仍继续筛查可得以下 comp.2 结果：

$$B = 0.584W_1 + 0.769W_3 - 0.240O$$

其累积贡献率达到 98.421%，最具相关性的因素为 10cm 湿度、100cm 湿度和土壤 C/N 比。

其所有线性组合结果累计贡献率如下图 7-3 所示：

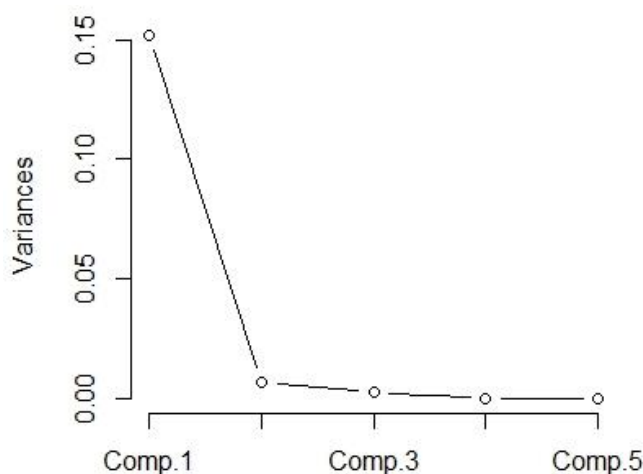


图 7-3 对照放牧强度下土壤板结化所有线性组合累计贡献率折线图

由图 7-3 观察所得 comp.2 得到的累计贡献率结果即可符合预期。

对于轻度放牧强度 LGI，通过主成分分析，从 5 个因子中选取主成分，得到 comp.1，带入土壤板结化表达式可得：

$$B = 0.176W_2 + 0.160W_3 - 0.381W_4 - 0.889W_5$$

其累积贡献率为 93.991%，可得预期结果，因此在对照组影响土壤板结化的主成分为 40cm 湿度、100cm 湿度、200cm 湿度和土壤 C/N 比。

若仍继续筛查可得以下 comp.2 结果：

$$B = 0.292W_1 - 0.193W_2 + 0.834W_3 - 0.321W_4 - 0.279O$$

其累积贡献率达到 97.290%，最具相关性的因素为 10cm 湿度、40cm 湿度、100cm 湿度、200cm 湿度和土壤 C/N 比。

其所有线性组合结果累计贡献率如下图 7-4 所示：

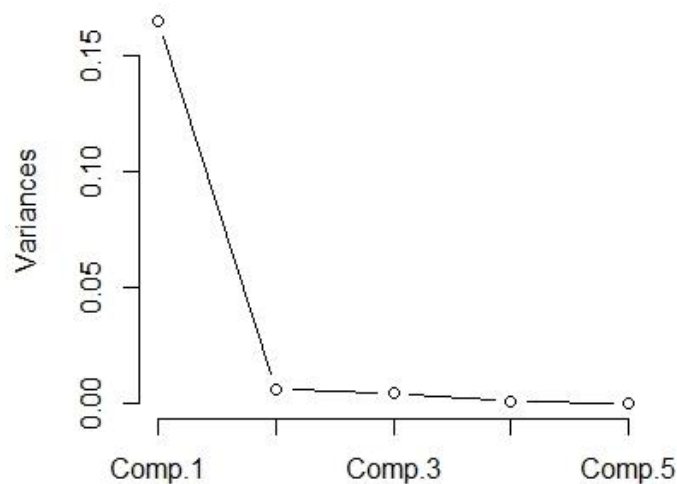


图 7-4 轻度放牧强度下土壤板结化所有线性组合累计贡献率折线图

由图 7-4 观察所得 comp.2 得到的累计贡献率结果即可符合预期。

对于中度放牧强度 MGI，通过主成分分析，从 5 个因子中选取主成分，得到 comp.1，带入土壤板结化表达式可得：

$$B = 0.116W_1 + 0.170W_2 + 0.192W_3 - 0.346W_4 - 0.895O$$

其累积贡献率为 86.895%，可得预期结果，因此在对照组影响土壤板结化的主成分为 10cm 湿度、40cm 湿度、100cm 湿度、200cm 湿度和土壤 C/N 比。

若仍继续筛查可得以下 comp.2 结果：

$$B = -0.208W_2 - 0.207W_3 + 0.862W_4 - 0.413O$$

其累积贡献率达到 96.202%，最具相关性的因素为 40cm 湿度、100cm 湿度、200cm 湿度和土壤 C/N 比。

其所有线性组合结果累计贡献率如下图 7-5 所示：

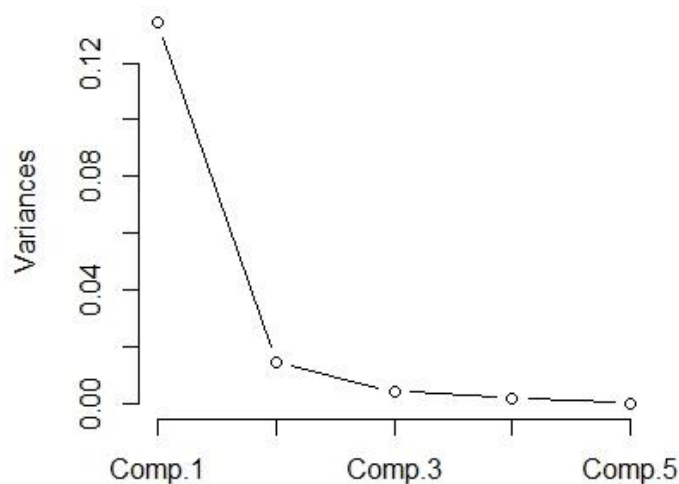


图 7-5 中度放牧强度下土壤板结化所有线性组合累计贡献率折线图

由图 7-5 观察所得 comp.2 得到的累计贡献率结果即可符合预期。

对于重度放牧强度 HGI，通过主成分分析，从 5 个因子中选取主成分，得到 comp.1，带入土壤板结化表达式可得：

$$B = 0.164W_2 + 0.143W_3 - 0.378W_4 - 0.897O$$

其累积贡献率为 94.678%，可得预期结果，因此在对照组影响土壤板结化的主成分为 40cm 湿度、100cm 湿度、200cm 湿度和土壤 C/N 比。

若仍继续筛查可得以下 comp.2 结果：

$$B = 0.468W_1 + 0.819W_3 - 0.216W_4 + 0.251O$$

其累积贡献率达到 98.337%，最具相关性的因素为 10cm 湿度、100cm 湿度、200cm 湿度和土壤 C/N 比。

其所有线性组合结果累计贡献率如下图 7-6 所示：

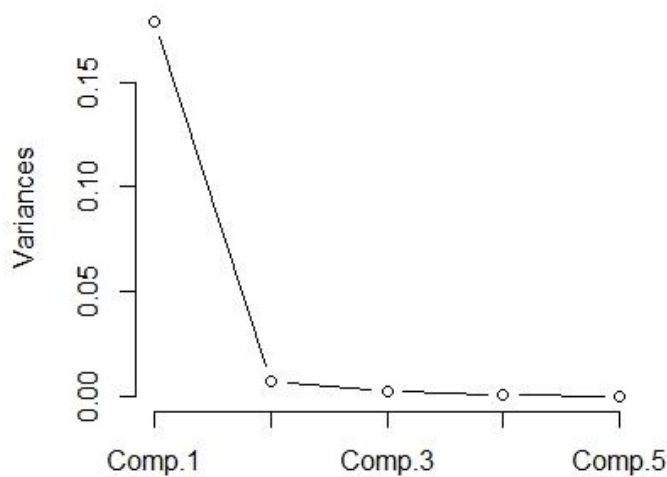


图 7-6 重度放牧强度下土壤板结化所有线性组合累计贡献率折线图

由图 7-6 观察所得 comp.2 得到的累计贡献率结果即可符合预期。

土壤化学性质和物理性质是影响土壤肥力重要因素，土壤化学性质包括：土壤有机碳 SOC、土壤无机碳 SIC、土壤全碳 STC、全 N、土壤 C/N 比等；土壤物理性质包括：土壤湿度、土壤容重等。一般来说，在保持土壤化学性质等基本不变情况下，降水会增加土壤湿度，而土壤湿度增加会提高草场植被覆盖率，在良好的植被覆盖情况下可以适当提高放牧强度，在一定区域内，放牧强度越高意味着更多的放牧数量，在不考虑放牧补贴和价格波动情况下，更多放牧数量代表更高的放牧收益。

7.3.3 2022 年土壤板结化程度预测

通过对土壤板结化影响因素在 2022 年的数据取平均，带入不同放牧强度下土壤板结化模型可得：

对照 NG：

$$\begin{aligned} B &= 0.584W_1 + 0.769W_3 - 0.240O \\ &= 0.584 \times 0.627 + 0.769 \times 0.316 - 0.240 \times 0.503 \\ &= 0.489 \end{aligned}$$

轻度放牧强度 LGI：

$$\begin{aligned}
B &= 0.292W_1 - 0.193W_2 + 0.834W_3 - 0.321W_4 - 0.279O \\
&= 0.292 \times 0.627 - 0.193 \times 0.37 + 0.834 \times 0.316 - 0.321 \times 0.692 - 0.279 \times 0.338 \\
&= 0.059
\end{aligned}$$

中度放牧强度 MGI:

$$\begin{aligned}
B &= -0.208W_2 - 0.207W_3 + 0.862W_4 - 0.413O \\
&= -0.208 \times 0.37 - 0.207 \times 0.316 + 0.862 \times 0.692 - 0.413 \times 0.337 \\
&= 0.315
\end{aligned}$$

重度放牧强度 HGI:

$$\begin{aligned}
B &= 0.468W_1 + 0.819W_3 - 0.216W_4 + 0.251O \\
&= 0.468 \times 0.627 + 0.819 \times 0.316 - 0.216 \times 0.692 + 0.251 \times 0.631 \\
&= 0.561
\end{aligned}$$

通过计算可得轻度放牧强度下土壤板结化程度最低，因此轻度放牧是最有利于草原土壤保护的放牧策略。

八、问题五模型建立与求解

8.1 数据处理

草原可持续发展，指的是土壤环境适宜。问题 5 所求的是在土壤没有板结化和沙漠化的情况下，草原可以放牧羊群的最大数量。建立目标规划模型。第一步，确立目标函数即最大的羊群数。第二步，写明限制条件：避免草原土壤的沙漠化或板结化到不可逆的情况下草原的最大羊群承受能力。第三步，找出影响羊群数量的三个因素：植被生物量(即干重)，放牧强度（对照 NG，轻度放牧 LGI，中度放牧 MGI，重度放牧 HGI）和湿度(土壤湿度)。第四步，找出三种因素与羊群数量的相关性并设表达式。第五步根据假设和数据进行模拟预测。有研究表明,高寒草甸的土壤全氮含量沿着放牧梯度呈下降趋势。所以，本题会用土壤含氮量来表示放牧强度。

8.2 模型建立

8.2.1 假设

- (1) 假设一块实验地的面积为 1 平方；1 平方千米草地一天能生产干草 93600Kg,其中可食用的为 40500Kg,
- (2) 假设不考虑土壤可持续发展的情况下，1 平方千米草地承载量为 260 只羊；
- (3) 假设一只羊的食量为 1.8Kg 草；
- (4) 对照（NG，0 羊/天/公顷）、轻度放牧强度（LGI，150 羊/天/公顷）、中度放牧强度（MGI，200 羊/天/公顷）和重度放牧强度（HGI，250 羊/天/公顷）。
- (5) 假设土壤沙漠化程度小于 0.4，土壤板结化程度小于 0.3 时即为土壤可持续发展状态。

8.2.2 具体模型

根据实际意义建立目标函数：

$$\begin{aligned} \max Z &= a_1x_1 + a_2x_2 + a_3x_3 \\ s.t. \begin{cases} b_1x_1 + b_2x_2 + b_3x_3 < 0.4 \\ c_1x_1 + c_2x_2 + c_3x_3 < 0.3 \end{cases} \end{aligned}$$

其中， Z 代表羊群数量， x_1 表示植被生物量， x_2 表示土壤含氮量， x_3 表示土壤湿度。 a_1, a_2, a_3 代表各个影响因素对羊群数量影响的权重，由于植被生物量增加会促进羊群数量增加，所以植被生物量与羊群数量呈正相关。随着放牧强度的增加，羊群数量会随着土壤含氮量的增加而减少，因此土壤含氮量与羊群数量呈负相关。土壤湿度增大，也会促进羊群数量增加，所以土壤湿度与羊群数量呈正相关。因此，参数 a_1, a_2, a_3 要满足 $a_1, a_3 > 0, a_2 < 0$ 。

下一步根据熵权法基本原理获取植被生物量、土壤含氮量和湿度权重系数：

Step1: 数据标准化处理

正向指标处理公式：

$$x'_{ij} = \frac{x_{ij} - \min\{x_{i1}, \dots, x_{im}\}}{\max\{x_{i1}, \dots, x_{im}\} - \min\{x_{i1}, \dots, x_{im}\}}$$

负向指标处理公式：

$$x'_{ij} = \frac{\max\{x_{i1}, \dots, x_{im}\} - x_{ij}}{\max\{x_{i1}, \dots, x_{im}\} - \min\{x_{i1}, \dots, x_{im}\}}$$

Step2: 计算各指标信息熵

信息熵的计算公式如下：

$$E_i = -\ln(m)^{-1} \sum_{j=1}^m p_{ij} \ln(p_{ij})$$

其中 $p_{ij} = \frac{x'_{ij}}{\sum_{j=1}^m x'_{ij}}$ ，若 $p_{ij} = 0$ ，则 $\lim p_{ij} \ln(p_{ij}) = 0$

Step3: 计算各指标权重

权重的计算公式如下：

$$W_i = \frac{1 - E_i}{\sum_{i=1}^k (1 - E_i)}$$

根据实际数据，由上式计算得到权重。

8.3 结果与讨论

由于数据缺失，不妨按假设情况处理。根据以上分析可得，在 300mm 降水的情况下，

草原土壤轻度沙漠化和板结化（可逆）时，草原最大承载羊群的数量是 210 只。

在 600mm 降水的情况下草原土壤轻度沙漠化和板结化（可逆）时，草原最大承载羊群的数量是 260 只。

在 900mm 降水的情况下，草原土壤轻度沙漠化和板结化（可逆）时，草原最大承载羊群的数量是 245 只。

在 1200mm 降水的情况下，草原土壤轻度沙漠化和板结化（可逆）时，草原最大承载羊群的数量是 190 只。

由此可得，放牧能够降低表层土壤湿度适度的放牧可以改善草原土壤质量、提高草原生物的多样性。羊群采食植物会减少植物的生物量，同时放牧也会改变植物的生长速率。过度放牧会导致单位公顷内的牲畜密度过大，草原植被结构破坏，地表裸露，水分流失，营养成分氮磷钾等元素缺失，土壤盐分随之增加，导致土壤盐碱化程度加重，最终造成草场退化、土地沙漠化。对植物进行采食时，一方面植物的地上生物量减少；另一方面，放牧对植物有刺激作用，改变了植物原有的生长速率，适当的放牧会刺激植物的超补偿生长，同样不合理的放牧也会降低植物的生长速率。

九、问题六模型建立与求解

9.1 数据处理

本题数据处理较之前题目相对简单，只需要在相同放牧策略和轻度放牧强度下，对 2012-2022 年土壤化学性质、土壤湿度和植被指数(NDVI)按时间排序，然后对对应年不同数值取平均。

9.2 模型建立

分别对以上处理数据运用 R 语言 ARIMA 模型，分别预测出 2023 年 9 月份数据即可。基本步骤类似于问题三，本题在此简化叙述过程。

9.3 结果与讨论

表 9-1 2023 年 9 月土地状态

土壤状态	分类	预测数值
土壤肥力	SOC 土壤有机碳	18.100
	SIC 土壤无机碳	4.820
	STC 土壤全碳	23.314
	全 N	2.250
	土壤 C/N 比	10.210
土壤湿度	10cm 湿度(kg/m2)	17.958
	40cm 湿度(kg/m2)	58.797
	100cm 湿度(kg/m2)	75.679
	200cm 湿度(kg/m2)	165.017
植被覆盖	植被指数(NDVI)	0.179

下面用图示或者动态演示方式分别预测示范区 2023 年 9 月土地状态：

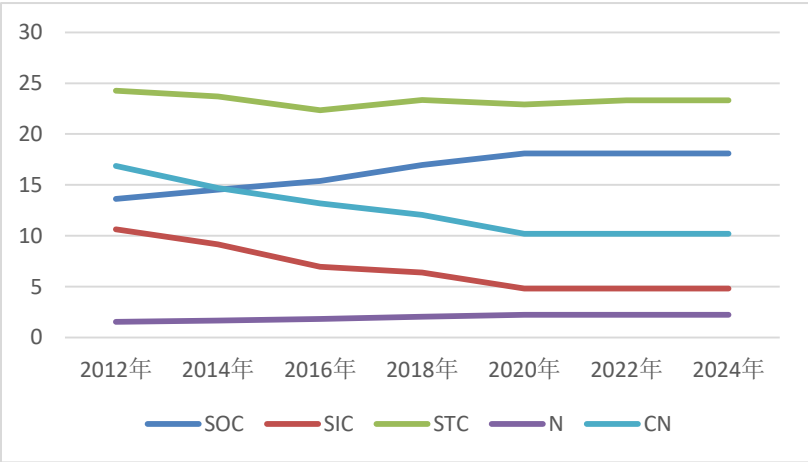


图 9-1 2012-2024 年土壤肥力情况

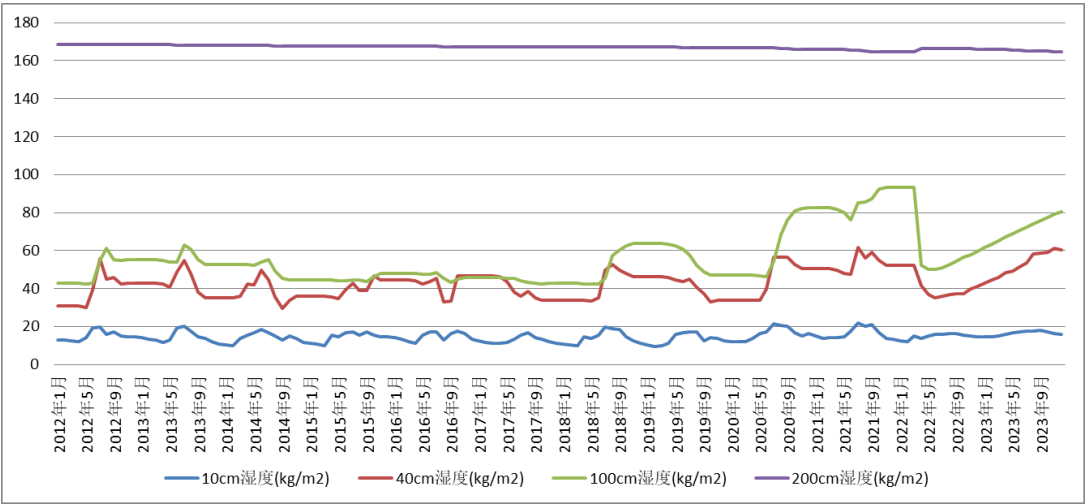


图 9-2 2012-2023 年土壤湿度

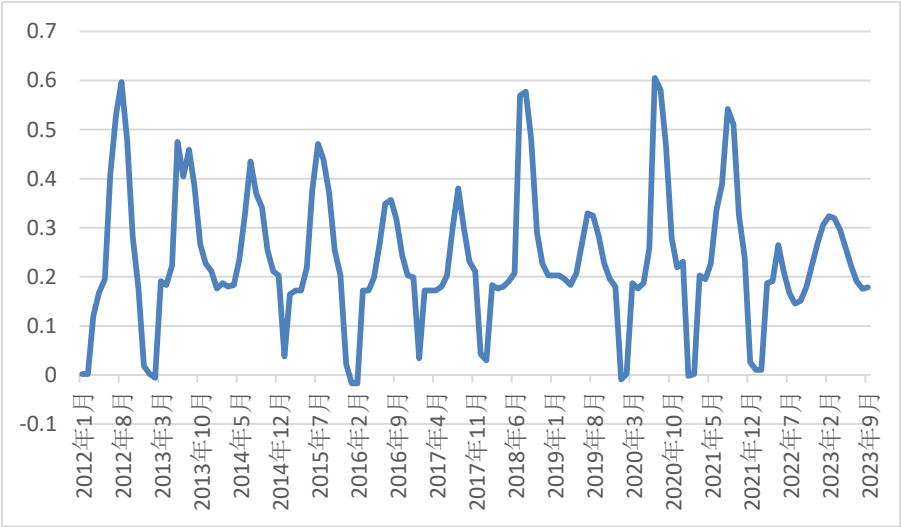


图 9-3 2012-2023 年植被指数

十、总结与评价

10.1 模型优点

(1) 本文通过运用统计及数学软件，例如 Python、R 语言、SPSS 等，尽可能使数据处理合理化，以得到更加准确的结果。

(2) 在数据的处理方面，定性数据定量化，定量数据标准化，通过查找大量文献尽可能使数据判定标准合理化。

(3) 运用多种时间序列方法对数据进行预测。长短期记忆 (LSTM) 时间序列模型、差分整合移动平均自回归模型—ARIMA 模型对 2022 年之后土壤数据进行了不同类型的预测。

(4) 建立多种方程形式以减小计算误差，例如微分方程、多项式方程、一维线性方程。运用主成分分析及皮尔逊相关系数，尽可能的使用较少的因素反应更好的变量。

(5) 尽可能绘制图形及表格来反映数据及模型精度，使得文章更加生动直观。

10.2 模型缺点及改进方向

(1) 数据较少，许多反应问题的数据没有找到而使得模型缺少变量。后续应广泛收集数据并与相关部门合作取得更加详细得数据，从而使得建立的模型考虑得更加全面。

(2) 模型方程建立多数采用一次变量，可能不能较好的拟合数据的真实情况。后续应建立真实数据拟合曲线来给自变量定阶。

(3) 数据清洗不够，可能会因数据极值而使得整体数据方差较大。应先对数据进行整体观察，筛除异常值以减小异常值对模型精度的影响。

(4) 由于时间原因，部分题目解答仅提供思路，对于细致的问题回答不到位。

(5) 本文中涉及代码简单重复较多，没有设计出简单直接的方以直接得到所需数据。如时间充裕，可以增强代码的一般化，使之可以解决一类问题。

参考文献

- [1] 奥蓝. “生态红线”这些年：发生在锡林郭勒草原上的牧业转型. <http://ku.m.chinanews.com/wapapp/zaker/sh/2020/11-26/9348187.shtml>. 2022.10.7.
- [2] 刘敦利. 基于栅格尺度的土地沙漠化预警模式研究[D]. 乌鲁木齐：新疆大学, 2010.
- [3] 张蕴薇, 韩建国, 李志强. 放牧强度对土壤物理性质的影响[J]. 草地学报, 2002(01): 74-78
- [4] Woodward S J R, Wake G C, McCall D G. Optimal grazing of a multi—paddock system using a discrete time model[J]. Agri—cultural Systems. 48(2): 119—139, 1995.
- [5] Bengio Y, Simard P, Frasconi P. Learning Long-Term Dependencies with Gradient Descent is Difficult[J]. IEEE Trans. Neural Netw. 5(2): 157—166, 1994.
- [6] Hochreiter S, Schmidhuber J J. Long short-term memory[J]. Neural Comput. 9(8): 1735-80, 1997.
- [7] Cheng M, Xu Q, Lv J, et al. MS-LSTM: A multi-scale LSTM model for BGP anomaly detection[J]. International Conference on the Network Protocols (ICNP), 2016;..
- [8] 夏丽. 基于 ARIMA 模型及回归分析的区域用电量预测方法研究[D]. 南京理工大学, 2013.
- [9] 张蒙, 李晓兵. 放牧对土壤有机碳的影响及相关过程研究进展[J], 草地学报, 26(2): 267-276, 2018.
- [10] 吴启堂主编, 环境土壤学, 北京: 中国农业出版社, 2015.
- [11] 林海明, 杜子芳. 主成分分析综合评价应该注意的问题[J]. 统计研究, 2013, 30(08): 25-31.

附录

第二问

```
## Multivariate-SingleStep-LSTM.py
```

```
from args import ms_args_parser
from util import train, test
```

```
LSTM_PATH = '../model/Multivariate-SingleStep-LSTM.pkl'
```

```
if __name__ == '__main__':
    args = ms_args_parser()
    flag = 'ms'
    train(args, LSTM_PATH, flag)
    test(args, LSTM_PATH, flag)
```

```
## args.py
```

```
import argparse
import torch
```

```
def ms_args_parser():
    parser = argparse.ArgumentParser()

    parser.add_argument('--epochs', type=int, default=30, help='input dimension')
    parser.add_argument('--input_size', type=int, default=6, help='input dimension')
    parser.add_argument('--output_size', type=int, default=21, help='output dimension')
    parser.add_argument('--hidden_size', type=int, default=64, help='hidden size')
    parser.add_argument('--num_layers', type=int, default=1, help='num layers')
    parser.add_argument('--lr', type=float, default=0.05, help='learning rate')
    parser.add_argument('--batch_size', type=int, default=1, help='batch size')
    parser.add_argument('--optimizer', type=str, default='adam', help='type of optimizer')
    parser.add_argument('--device', default=torch.device("cuda" if torch.cuda.is_available() else
"cpu"))
    parser.add_argument('--weight_decay', type=float, default=1e-4, help='weight decay')
    parser.add_argument('--bidirectional', type=bool, default=False, help='LSTM direction')

    args = parser.parse_args()

    return args
```

```

## data_process.py

import os
import random

import numpy as np
import pandas as pd
import torch
from torch.utils.data import Dataset, DataLoader

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

def setup_seed(seed):
    os.environ['PYTHONHASHSEED'] = str(seed)
    torch.manual_seed(seed)
    torch.cuda.manual_seed_all(seed)
    np.random.seed(seed)
    random.seed(seed)
    torch.backends.cudnn.deterministic = True

def load_data():
    path = os.path.dirname(os.path.realpath(__file__)) + '/data/q2(200).csv'
    df = pd.read_csv(path, encoding='gbk')
    columns = df.columns
    df.fillna(df.mean(), inplace=True)
    MAX = np.max(df[columns[0]])
    MIN = np.min(df[columns[0]])
    df[columns[0]] = (df[columns[0]] - MIN) / (MAX - MIN)

    return df, MAX, MIN

class MyDataset(Dataset):
    def __init__(self, data):
        self.data = data

    def __getitem__(self, item):
        return self.data[item]

    def __len__(self):
        return len(self.data)

def nn_seq_ms(B):
    print('data processing...')

```

```

data, m, n = load_data()
load = data[data.columns[0]]
load = load.tolist()
data = data.values.tolist()
seq = []
for i in range(len(data) - 50 - 21):
    train_seq = []
    train_label = []
    for j in range(i, i + 50):
        x = [load[j]]
        for c in range(1, 6):
            x.append(data[j][c])
        train_seq.append(x)
    train_label.append(load[i + 50: i + 71])
    train_seq = torch.FloatTensor(train_seq)
    train_label = torch.FloatTensor(train_label).view(-1)
    seq.append((train_seq, train_label))
Dtr = seq[0:int(len(seq))]

seq = []
i = len(data) - 50
train_seq = []
for j in range(i, i + 50):
    x = [load[j]]
    for c in range(1, 6):
        x.append(data[j][c])
    train_seq.append(x)
train_seq = torch.FloatTensor(train_seq)
seq.append(train_seq)
Dte = seq[0:int(len(seq))]

# Dtr = seq[0:int(len(seq) * 0.7)]
# Dte = seq[int(len(seq) * 0.7):len(seq)]

# train_len = int(len(Dtr) / B) * B
# test_len = int(len(Dte) / B) * B
# Dtr, Dte = Dtr[:train_len], Dte[:test_len]

train = MyDataset(Dtr)
test = MyDataset(Dte)

Dtr = DataLoader(dataset=train, batch_size=B, shuffle=False, num_workers=0)
Dte = DataLoader(dataset=test, batch_size=B, shuffle=False, num_workers=0)

```

```

    return Dtr, Dte, m, n

def get_mape(x, y):
    """
    :param x:true
    :param y:pred
    :return:MAPE
    """
    return np.mean(np.abs((x - y) / x))

## models.py

from torch import nn
import torch
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

class LSTM(nn.Module):
    def __init__(self, input_size, hidden_size, num_layers, output_size, batch_size):
        super().__init__()
        self.input_size = input_size
        self.hidden_size = hidden_size
        self.num_layers = num_layers
        self.output_size = output_size
        self.num_directions = 1
        self.batch_size = batch_size
        self.lstm = nn.LSTM(self.input_size, self.hidden_size, self.num_layers,
batch_first=True)
        self.linear = nn.Linear(self.hidden_size, self.output_size)

    def forward(self, input_seq):
        h_0 = torch.randn(self.num_directions * self.num_layers, self.batch_size,
self.hidden_size).to(device)
        c_0 = torch.randn(self.num_directions * self.num_layers, self.batch_size,
self.hidden_size).to(device)
        # print(input_seq.size())
        seq_len = input_seq.shape[1]
        # seq_len = 21
        # input(batch_size, seq_len, input_size)
        input_seq = input_seq.view(self.batch_size, seq_len, self.input_size)
        # output(batch_size, seq_len, num_directions * hidden_size)
        output, _ = self.lstm(input_seq, (h_0, c_0))
        # print('output.size=', output.size())
        # print(self.batch_size * seq_len, self.hidden_size)

```

```

        output = output.contiguous().view(self.batch_size * seq_len, self.hidden_size) # (5 *
30, 64)
        pred = self.linear(output) # pred()
        # print('pred=', pred.shape)
        pred = pred.view(self.batch_size, seq_len, -1)
        pred = pred[:, -1, :]
        return pred

```

util.py

```

from itertools import chain

```

```

import torch
from scipy.interpolate import make_interp_spline
from torch import nn
import numpy as np
import matplotlib.pyplot as plt
from models import LSTM, BiLSTM
from data_process import nn_seq, nn_seq_ms, nn_seq_mm, device, get_mape, setup_seed
setup_seed(20)

```

```

def train(args, path, flag):

```

```

    if flag == 'us':
        Dtr, Dte, m, n = nn_seq(B=args.batch_size)
    elif flag == 'ms':
        Dtr, Dte, m, n = nn_seq_ms(B=args.batch_size)
    else:
        Dtr, Dte, m, n = nn_seq_mm(B=args.batch_size, num=args.output_size)

```

```

    input_size, hidden_size, num_layers = args.input_size, args.hidden_size, args.num_layers
    output_size = args.output_size
    if args.bidirectional:
        model = BiLSTM(input_size, hidden_size, num_layers, output_size,
batch_size=args.batch_size).to(device)
    else:
        model = LSTM(input_size, hidden_size, num_layers, output_size,
batch_size=args.batch_size).to(device)

```

```

    loss_function = nn.MSELoss().to(device)
    if args.optimizer == 'adam':
        optimizer = torch.optim.Adam(model.parameters(), lr=args.lr,
weight_decay=args.weight_decay)

```



```

else:
    optimizer = torch.optim.SGD(model.parameters(), lr=args.lr,
                                momentum=0.9, weight_decay=args.weight_decay)

# training
loss = 0
for i in range(args.epochs):
    cnt = 0
    for (seq, label) in Dtr:
        cnt += 1
        seq = seq.to(device)
        label = label.to(device)
        y_pred = model(seq)
        loss = loss_function(y_pred, label)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
        # if cnt % 100 == 0:
        #     print('epoch', i, ':', cnt - 100, '~', cnt, loss.item())
    print('epoch', i, ':', loss.item())

state = {'model': model.state_dict(), 'optimizer': optimizer.state_dict()}
torch.save(state, path)

def test(args, path, flag):
    if flag == 'us':
        Dtr, Dte, m, n = nn_seq(B=args.batch_size)
    elif flag == 'ms':
        Dtr, Dte, m, n = nn_seq_ms(B=args.batch_size)
    else:
        Dtr, Dte, m, n = nn_seq_mm(B=args.batch_size, num=args.output_size)
    pred = []
    y = []
    print('loading model...')
    input_size, hidden_size, num_layers = args.input_size, args.hidden_size, args.num_layers
    output_size = args.output_size
    if args.bidirectional:
        model = BiLSTM(input_size, hidden_size, num_layers, output_size,
                        batch_size=args.batch_size).to(device)
    else:
        model = LSTM(input_size, hidden_size, num_layers, output_size,
                     batch_size=args.batch_size).to(device)
    # model = LSTM(input_size, hidden_size, num_layers, output_size,
    #               batch_size=args.batch_size).to(device)

```

```

model.load_state_dict(torch.load(path)['model'])
model.eval()
print('predicting...')
for (seq, target) in Dte:
    target = list(chain.from_iterable(target.data.tolist()))
    y.extend(target)
    seq = seq.to(device)
    with torch.no_grad():
        y_pred = model(seq)
        y_pred = list(chain.from_iterable(y_pred.data.tolist()))
        pred.extend(y_pred)

for seq in Dte:
    seq = seq.to(device)
    with torch.no_grad():
        y_pred = model(seq)
        y_pred = list(chain.from_iterable(y_pred.data.tolist()))
        pred.extend(y_pred)

y, pred = np.array([y]), np.array([pred])
# pred = np.array([pred])
y = (m - n) * y + n
pred = (m - n) * pred + n
print(pred)
print('mape:', get_mape(y, pred))
# plot
x = [i for i in range(len(y[0]))]
# x_smooth = np.linspace(np.min(x), np.max(x), 900)
# y_smooth = make_interp_spline(x, y)(x_smooth)
plt.plot(x, y[0][0:len(y[0])], c='green', marker='*', ms=1, alpha=0.75, label='true')
# y_smooth = make_interp_spline(x, pred.T[150:300])(x_smooth)
plt.plot(x, pred[0][0:len(pred[0])], c='red', marker='o', ms=1, alpha=0.75, label='pred')
plt.grid(axis='y')
plt.legend()
plt.show()

```

第三问

```

aSOC<-read.table(file="C:/Users/DELL/Desktop/NG-G17.csv",sep="," ,header=T)
xSOC<-ts(aSOC$SOC,start=2012)
auto.arima(xSOC)
SOC.fit<-arima(xSOC,order=c(0,0,0))
SOC.fore<-forecast(SOC.fit,h=1)
SOC.fore

```

```

aSIC<-read.table(file="C:/Users/DELL/Desktop/NG-G17.csv",sep="," ,header=T)
xSIC<-ts(aSIC$SIC,start=2012)
auto.arima(xSIC)
xSIC.fit<-arima(xSIC,order=c(0,0,0))
xSIC.fore<-forecast(xSIC.fit,h=1)
xSIC.fore

```

```

aSTC<-read.table(file="C:/Users/DELL/Desktop/NG-G17.csv",sep="," ,header=T)
xSTC<-ts(aSTC$STC,start=2012)
auto.arima(xSTC)
xSTC.fit<-arima(xSTC,order=c(0,0,0))
xSTC.fore<-forecast(xSTC.fit,h=1)
xSTC.fore

```

```

aN<-read.table(file="C:/Users/DELL/Desktop/NG-G17.csv",sep="," ,header=T)
xN<-ts(aN$N,start=2012)
auto.arima(xN)
xN.fit<-arima(xN,order=c(0,0,0))
xN.fore<-forecast(xN.fit,h=1)
xN.fore

```

```

aCN<-read.table(file="C:/Users/DELL/Desktop/NG-G17.csv",sep="," ,header=T)
xCN<-ts(aCN$CN,start=2012)
auto.arima(xCN)
xCN.fit<-arima(xCN,order=c(0,0,0))
xCN.fore<-forecast(xCN.fit,h=1)
xCN.fore

```

第四问

```

a<-read.table(file="C:/Users/DELL/Desktop/主成分分析.csv",sep="," ,header=T)
x1<-ts(a$x1)
x2<-ts(a$x2)
x3<-ts(a$x3)
x4<-ts(a$x4)
x5<-ts(a$x5)
x6<-ts(a$x6)
X<-data.frame(x1,x2,x3,x4,x5,x6)
test.pr<-princomp(X,cor=FALSE)
summary(test.pr,loadings=TRUE) #做主成分分析
plot(test.pr,type="line") #画主成分碎石图

```

```

a<-read.table(file="C:/Users/DELL/Desktop/NG.csv",sep="," ,header=T)
x1<-ts(a$sd10)
x2<-ts(a$sd40)

```

```

x3<-ts(a$sd100)
x4<-ts(a$sd200)
x5<-ts(a$CN)
X<-data.frame(x1,x2,x3,x4,x5)
test.pr<-princomp(X)
summary(test.pr,loadings=TRUE) #做主成分分析
plot(test.pr,type="line") #画主成分碎石图

```

第六问

```

aSOC<-read.table(file="C:/Users/DELL/Desktop/6.csv",sep=" ",header=T)
xSOC<-ts(aSOC$SOC)
auto.arima(xSOC)
SOC.fit<-arima(xSOC,order=c(0,1,0))
SOC.fore<-forecast(SOC.fit,h=2)
SOC.fore

```

```

aSIC<-read.table(file="C:/Users/DELL/Desktop/6.csv",sep=" ",header=T)
xSIC<-ts(aSIC$SIC)
auto.arima(xSIC)
xSIC.fit<-arima(xSIC,order=c(0,1,0))
xSIC.fore<-forecast(xSIC.fit,h=2)
xSIC.fore

```

```

aSTC<-read.table(file="C:/Users/DELL/Desktop/6.csv",sep=" ",header=T)
xSTC<-ts(aSTC$STC)
auto.arima(xSTC)
xSTC.fit<-arima(xSTC,order=c(0,0,0))
xSTC.fore<-forecast(xSTC.fit,h=2)
xSTC.fore

```

```

aN<-read.table(file="C:/Users/DELL/Desktop/6.csv",sep=" ",header=T)
xN<-ts(aN$N)
auto.arima(xN)
xN.fit<-arima(xN,order=c(0,1,0))
xN.fore<-forecast(xN.fit,h=2)
xN.fore

```

```

aCN<-read.table(file="C:/Users/DELL/Desktop/6.csv",sep=" ",header=T)
xCN<-ts(aCN$CN)
auto.arima(xCN)
xCN.fit<-arima(xCN,order=c(0,1,0))
xCN.fore<-forecast(xCN.fit,h=2)
xCN.fore

```

```
aNDVI<-read.table(file="C:/Users/DELL/Desktop/6.csv",sep=";",header=T)
xNDVI<-ts(aNDVI$NDVI)
auto.arima(xNDVI)
xNDVI.fit<-arima(xNDVI,order=c(2,0,2))
xNDVI.fore<-forecast(xNDVI.fit,h=17)
xNDVI.fore
```