《MATLAB 程序设计》大作业个人汇报

COVID-19 疫情数据的简要统计分析与预测*

王逸扬（19300180016）
杨耕智（19300180112）

2022 年 5 月 21 日

*相关代码及其版本演进托管在https://github.com/maix00/StatisticsAnalysis.

目录

1 整体分析思路与代码分工

众所周知，2019 年底开始的 COVID-19 疫情给人们的生产生活造成了极大的影响，如何利用疫情相关数据、如何通过数据挖掘、分析、预测，以指导人们作出更加有利于社会发展的决策，是当今较为热门的研究方向. 在此，我们将根据所给的有关数据，尽可能地贴合这个目的，做出一些简要的分析与预测.

1.1 数据观察与简要分析

首先，观察所给数据. 有两个数据集，一个是 `country.csv`（以下称「国家数据」），另一个是 `daily_info.csv`（以下称「每日数据」）. 每日数据中包括 21 个国家的有关数据，其中 Senegal 在国家数据中有数据缺失，遂只对余下 20 个国家进行分析，分别是 Australia, Canada, Chile, China, Egypt, Ethiopia, France, Germany, Hungary, Iceland, Japan, Malaysia, Saudi Arabia, South Africa, South Korea, Sweden, Thailand, United States, 以及 Zimbabwe.

这 20 个国家的国家数据中，每个国家都有 12 个特征，简要分类：

地理位置: 所属大洲 `continent`;

人口成分: 人口 `population`, 每平方千米人口密度 `population_density`, 中位年龄 `median_age`, 65 岁以上人口占比 `aged_65_older`, 70 岁以上人口占比 `aged_70_older`;

经济环境: 按购买力平价的人均国民生产总值 `gdp_per_capita`;

卫生水平: 男性吸烟率 `male_smokers`, 女性吸烟率 `female_smokers`, 每年每十万人心血管疾病死亡率 `cardiovasc_death_rate`, 每千人床位数 `hospital_beds_per_thousands`, 预期寿命 `life_ expectancy`.

据此，我们可以对这些国家进行无监督聚类分析，在每一类中选取一个代表性的国家构建一分析预测模型（利用每日数据和/或国家特征），并对同一类中的其他国家验证该模型的有效

性. 但是，需要指出，因为所给数据的有限性，所建立或选择的模型是有限的，这里的特征作为模型参数的个数也是有限的，因此这里的多数特征在本文中只能用于聚类.

对于每日数据，我们首先指出，在数据的导入与缺失值处理上，这个数据集遇到了一些困难，对于这些问题，我们将在节2与节3中分别尝试解决. 下面，对每日数据的各字段（除所属大洲与时间戳外），简要分类：

严重程度: 新增与累计病例:new_cases, total_cases; 住院与重症监护 ICU 人数:hosp_patients, icu_patients; 近七日平均检测阳性率: positive_rate;

检测能力: 新增与累计检测人次:new_tests, total_tests; 近七日平均检测阳性率:positive_rate;

疫苗接种: 新增与累计疫苗接种人次: new_vaccinations, total_vaccinations;

管控力度: 政府管控力度: stringency_index.

这些时间序列数据，(1) 可以用来构建一分析预测模型，(2) 也可以取某一时间段内的最大、平均等统计指标（时间序列特征），用来对不同国家进行监督或无监督聚类分析. 这里的聚类分析，得到的是不同国家的不同疫情发展模式，与前面的利用国家数据得到的关于国家特性的聚类不同.

另外，需要指出，因为所给数据的有限性，所能建立或选择的分析预测模型是极其有限的，比如，由于缺少治愈、死亡数据，不能采用传统的传染病模型 SIR 至 SEIR 等，又比如，由于缺少流调数据，无法统计计算基本再生数 R_0、潜伏期平均长度 $\bar{T_L}$ 等与传染病有关的统计指标. 此外，正因为所能建立或选择的模型是有限的，每日数据中的一些字段在本文中可能不会被用到.

1.2 整体分析思路

综合上面的简要分析，我们将本文的整体分析思路概括如下.

第一种路径 根据国家特征进行聚类，而后利用代表国家的每日数据构造一分析预测模型.

第一步:（聚类分析）对 20 个国家的特征数据，先进行主成分分析，选取前 5 个主特征后进行 5-mean 聚类分析.

第二步:（构建模型）基于 LSTM 神经网络模型，对代表国家一定时期内的每日新增病例数据进行学习.

第三步:（预测检验）利用上一步得到的模型，对时期外的数据进行预测检验，并对同一类中的其他国家的数据进行预测检验.

此路径能得到一种基于 LSTM 神经网络的分析预测模型.

第二种路径 根据每日数据和/或国家数据构造时间序列特征，而后进行谱系聚类.

第一步:（构造特征）

第二步:（聚类分析）

此路径能对不同国家的疫情发展模式进行分类，针对分类结果可以提出更有针对性的抑制疫情发展的建议. https://www.it610.com/article/1515470890224123904.htm.

第三种路径 拟合部分国家疫情发展初期缺失的数据.
https://www.it610.com/article/1515470890224123904.htm.

1.3 代码分工

因为疫情的关系，我们两位同学，不得不线上联系，我们也因此第一次尝试使用 github 平台. 仓库网址为https://github.com/maix00/StatisticsAnalysis. 有关代码分工的详细情况，在平台上可以清晰看到，这里就作简单介绍.

节2与节3数据导入、缺失值处理的有关代码主要是由王逸扬同学完成的，节4数据分析的有关代码主要是由杨耕智同学完成的.

2 数据的导入与检索

导入数据已有可以使用的 `readtable` 函数，且可以通过 `detectImportOptions` 函数，在导入数据前，预先探测并修改导入参数. 在此过程中，我们发现如下问题:

1. 以每日数据的 `new_vaccinations` 与 `total_vaccinations` 变量为例. 由于疫情发生初期长时间没有此类数据，`detectImportOptions` 函数探测认为，此变量的变量类型是为 `char`，因此需要手动将导入参数修改为 `double`.
2. 在导入每日数据时，通常最后需要检索出某个国家、某个时间段的数据，目的是形成时序数据. 为此，MATLAB 可以通过括号索引检索，但是不同类型的数据检索的方式不同，不是很便利，没有专门的检索函数可以使用.

为了解决上面发现的问题，并寄希望于能在单一函数中解决全部的导入与检索问题，我们利用了 MATLAB 的面向对象编程，编写了一个类 `StatisticsAnalysis`（之所以取这个名字，是因为之后还将赋予它更多的功能）. 下面用一个例子来阐述这个类的作用. 比如，我们想要导入法国在 2020 年的新增与累计接种疫苗人次——

```
daily_SA = StatisticsAnalysis( ...
    'TablePath', './data/COVID19/daily_info.csv', ...
    'ImportOptions', { ...
        'VariableTypes', { ...
            'new_vaccinations', 'double', ...
            'total_vaccinations', 'double' ...
            }, ...
        'SelectedVariableNames', ...
            {'date', 'new_vaccinations', 'total_vaccinations'} ...
        }, ...
    'SelectTableOptions', { ...
        'location', 'France', ...
        'date', timerange("2020-01-01", "2020-12-31", 'closed') ...
        } ...
    );
daily = daily_SA.TimeTable;
```

其中，`"TablePath"`、`"ImportOptions"` 和`"SelectTableOptions"` 是函数参数名，`TimeTable` 是该类的一个属性. 如果我们还想导入法国在 2020 年的新增与累计病例数，我们只需——

```
daily2 = daily_SA.Update('ImportOptions', { ...
        'SelectedVariableNames', {'date', 'new_cases', 'total_cases'} ...
        }...
    ).TimeTable;
```

上面的方法不会重复导入、检索数据. 更多关于这个类的信息，请参见附录A.

3 数据的缺失值处理

以每日数据为例，我们发现如下问题：

1. 在疫情发展初期，一些变量有很多缺失值，可以删除行，一些情况下也可以填充为 0.
2. 多个变量体现出了「新增-累计」的特征，如每日新增病例与累计病例、每日新增接种人次与累计接种人次，这些数据会有意外的缺失值，分为以下几种情况：
   (1) 数值意外地未被记录，但是可以从其周围的数据中恢复；
   (2) 某日的新增数据为零，与近几日数据不相符合，可以认为是离群值；这种情况认为是统计滞后引起的，可以通过近几日的平均来进行光滑，也可以不作改动.
   (3) 某日的总量数据比前一日乃至前几日的总量数据要少，使得当日的新增数据被记录为缺失值；这种情况认为是统计更正引起的，由于不知道这些多被记录的数据的分布情况，不能准确地作出修改.
   我们认为此种情况可以有多种处理方式，这里列举两种：(1-) 根据模型的选择，可以不作修改，直接将当日新增数据记录为负值；(2-) 选取窗口进行光滑，用近几日数据的平均等统计数字记录当日的新增数据，计算前一日至当日的差值，指数衰减地分配到当日的前几日之前. （之所以使用衰减的分配，我们隐含了一种假设，即数据更正多是发生在案例数快速增长的时期，此时可能因为统计口径的原因多记录了一些案例，这些案例应当更可能分布在时间较近的时刻. 之所以分配到几日之前，我们隐含假设统计和统计更正本身需要时间.）

为了解决这些问题，我们编写了 `TableMissingValues.m`，可以用来分析缺失值的分布，在这之后再经过人工判断，可以选择不同的参数，以恢复数据. 我们也将这个功能整合到了类 `StatisticsAnalysis` 中. 下面举一个例子来阐述其用法. 例如对法国在 2020 年的新增与累计病例数进行缺失值处理，我们只需在上文的基础上——

```
daily_SA.MissingValuesReport
```

```
ans =
        Map: [343x3 logical]
       date: {}
  new_cases: {[72 72] [75 75] [91 91] [97 97] [122 122] [131 132] [157 157] [286 286]}
total_cases: {}
```

可见数据中只有新增数据缺失，接着——

```
MissingValuesOptions = { {'new_cases', 'total_cases'}, 'Increment-Addition', ...
        {'InterpolationStyle','LinearRound', ...
        'RemoveFirstRows',false,'RemoveLastRows',false}
    };
```

```
daily2 = daily_SA.Update('MissingValuesOptions', MVO).TimeTable;
```

这里缺省使用了上文所述的指数衰减，因为其总量数据不单调. 更多的缺失值信息可以通过下面的方法获取——

```
daily_SA.MissingValuesReport.increment_addition_new_cases_total_cases
```

```
ans =
          Increment: 'new_cases'
           Addition: 'total_cases'
     IncrementWhere: [0 1 0]
      AdditionWhere: [0 0 1]
IncrementMissingMap: [343x1 logical]
 AdditionMissingMap: [343x1 logical]
 DecreasingAddition: {[71 72] [74 75] [90 91] [96 97] [121 122] [130 131]
                      [131 132] [156 157] [285 286]}
      MissingBlocks: [8x1 struct]
    tpMissingBlocks: [8x1 struct]
 MissingBlocksGroup: [8x1 struct]
```

另外，自定义的插值、衰减函数可以通过参数用函数句柄导入. 更多的信息请参见附录A.

4 数据分析过程

4.1 一般的统计指标

4.2 聚类分析

4.3 疫情数据的分析预测

A StatisticsAnalysis 类

详见 github 仓库，网址https://github.com/maix00/StatisticsAnalysis. 本类存储在./functions/@StatisticsAnalysis. 本类目前有如下功能：

1. **修改导入参数，并导入表格**；

   修改导入参数，请使用 ImportOptions 参数，类型可以是 struct 或采用 name/value-pair 的 cell；name 应当是你所要修改的 detectImportOptions 中得到的属性.

   另外，如果有设置多个导入参数的需要，请将值置于 1x1 cell，比如导入第 3 ~ 10 及 20 ~ 30 行，则为 DataLines:{{[3 10] [20 30]}}.

2. **检索表格**；

   传入检索参数，请使用 SelectTableOptions，类型可以是 struct 或采用 name/value-pair 的 cell；name 应当是你所要检索的变量名.

   导入与检索后的表格，可以通过属性名 Table 或 TimeTable 访问. 访问原表格请使用属性名 WholeTable.

3. **缺失值探测、修正**；

   传入修正缺失值的参数，请使用 MissingValuesOptions，类型可以是 struct 或 N_by_3 cell. 传入 cell 时，每行第一位是涉及的变量名 VariableNames，第二位是使用的修正方法 Style，第三位是该修正方法所需要的其他参数，类型可以是 struct 或者采用 name/value-pair 的 cell.

   查看探测信息，请使用属性名 MissingValuesReport. 缺失值修正后的表格将覆盖掉 obj.Table. 此外，允许的参数值如下表所示：

修正方法	参数名	允许参数值
MissingDetect	——	——
Increment-Addition	RemoveLastRows	缺省 true
	ConstantValues_FirstRows	缺省空
	RemoveFirstRows	缺省 true
	InterpolationStyle InterpolationFunction InterpolationStyle_P InterpolationFunction_P InterpolationStyle_C InterpolationFunction_C	Style 缺省"Linear" 或 "LinearRound"，允许函数名字符串如"spline"；Function 允许函数句柄.
	DecreasingAdditionStyle DecreasingAdditionParameters	Style 缺省"Exponential"，允许 "LinearScale", "DoNothing".
Interpolation	InterpolationStyle InterpolationFunction	
ConstantValues	ConstantValues	缺省空

其中，函数句柄句法、相关参数有

(1-) T = InterpolationFunction(T, startRow, endRow, VariableMap);

(2-) `T = InterpolationFunction_C(T, startRow, endRow, IncrementWhere, AdditionWhere)`; 注：新增-累计类数据会有两种需要插值的情况，记为 P 与 C.

(3-) 非单调累计数据处理方法`"Exponential"` 的参数较多，包括 RoundingWindowAhead, RoundingWindowBehind, RoundingStrategy, RoundingScale, ExponentialRate, AcceptedRatioMinimum, AcceptedRatioMaximum, SpanAheadSkip.

4. **给不同变量贴上不同标签，并对不同标签的变量，计算传入或预设的统计指标，并将计算结果添加到 `Table.Properties.CustomProperties`.**

   传入标签与统计指标的参数，请使用 `TagsGenerateOptions`，类型可以是 `struct` 或 `cell`；`struct` 的域名，或者 `cell` 的左列，应当是函数`./function/@StatisticsAnalysis/TagsGenerate` 的参数名，包括

   - `CustomTagName`，例如 `{"continuous", [0 1 1]}`;
   - `CustomTagFunction`, 例如 `{"continuous", "variance", @(x,y)tsnanvar(x)}`, 其中 `x` 和 `y` 分别是变量所在列全体，及其去重无缺全体.

注：检索功能引用了`./function/selecttable.m` 与 MATLAB 内置类 `timerange`，缺失值修正引用了`./function/@TableMissingValues`. 更新各种参数可以使用类方法 `Update`.

B 利用有关数据可以进行的可视化