

# BSTVC R 包使用指南

宋超

唐先腾

2025-01-28

## 目录

|          |                              |           |
|----------|------------------------------|-----------|
| <b>1</b> | <b>R 包介绍</b>                 | <b>2</b>  |
| 1.1      | 理论基础 . . . . .               | 2         |
| 1.2      | BSTVC 方法优势 . . . . .         | 3         |
| 1.3      | R 包功能特点 . . . . .            | 3         |
| <b>2</b> | <b>R 包安装</b>                 | <b>4</b>  |
| 2.1      | 从 GitHub 上安装包 . . . . .      | 4         |
| 2.2      | 安装依赖包 . . . . .              | 4         |
| 2.3      | 加载 <i>BSTVC</i> 包 . . . . .  | 4         |
| <b>3</b> | <b>利用 BSTVC 函数构建局域时空回归模型</b> | <b>5</b>  |
| 3.1      | 导入时空面板数据与检查 . . . . .        | 5         |
| 3.2      | 解释变量预处理 . . . . .            | 6         |
| 3.3      | BSTVC 建模 . . . . .           | 7         |
| 3.4      | 模型结果 . . . . .               | 8         |
| 3.5      | BSTVC 函数的输出及其可视化 . . . . .   | 9         |
| <b>4</b> | <b>利用 BSVC 函数构建局域空间回归模型</b>  | <b>22</b> |
| 4.1      | 导入空间截面数据与检查 . . . . .        | 22        |
| 4.2      | 解释变量预处理 . . . . .            | 23        |
| 4.3      | BSVC 建模（自定义空间权重矩阵） . . . . . | 24        |
| 4.4      | 模型结果 . . . . .               | 26        |
| 4.5      | BSVC 函数的输出及其可视化 . . . . .    | 26        |

|               |    |
|---------------|----|
| 1 R 包介绍       | 2  |
| 5 BSTVC 的引用文献 | 36 |
| 6 应用案例        | 36 |
| 7 Q & A       | 37 |
| 8 版本更新与技术支持   | 38 |

## 1 R 包介绍



# BAYESIAN STVC

**BSTVC R 包：时空异质视角下影响因素分析、关键驱动因素识别与动态预测的“全地图”建模工具。**

BSTVC 包提供了一个完整统一的“全地图”地理建模框架来精准捕捉具有时空差异的变量关系，旨在揭示多源解释变量对目标变量的时空异质影响机制，即时空非平稳效应。该包简洁易用，既能满足专业人士的深度需求，又降低了贝叶斯复杂建模的门槛，使更广泛的用户群体能够轻松应用先进的贝叶斯局域时空回归分析方法，解析和解释复杂的时空面板数据。适用于涉及地理时空数据分析的各类自然与人文科学领域，包括但不限于公共卫生、医学地理、环境健康、卫生经济和社会医学等学科（Song and Tang, 2025）。

### 1.1 理论基础

时空异质耦合分析同时考虑了基于两个地理学定律的时空异质性和时空自相关，并在同一框架内结合时间和空间维度进行耦合分析。这种方法是当前地理数据分析中最全面、精准且具有最强证据力的分析理念，远超传统的非空间分析、单一空间或时间分析，以及时空分离分析方法。BSTVC R 包作为时空异质耦合分析的前沿最新工具，其统计机理如下。

贝叶斯时空变系数（Bayesian Spatiotemporally Varying Coefficients, BSTVC）模型是一类基于贝叶斯统计内核的局域时空回归分析方法（[Song et al., 2019; 2020; 2022](#)），其显著优势在于，利用“全地图”单独建模框架对所有局部回归系数的时空变化进行统一拟合，从而能够精准捕捉解释变量对目标变量的时空异质性影响，即揭示时空非平稳性。BSTVC 系列模型（目前包括 STVI、STIVI、STVC、STIVC 四类子模型）为揭示目标变量的复杂时空动态变化和时空影响机制提供了强有力的工具（[Song et al., 2022](#)）。

贝叶斯空间变系数（Bayesian Spatially Varying Coefficients, BSVC）模型是 BSTVC 模型的空间维度精简版，仅用于识别具有空间异质性的变量关系，即空间非平稳效应。其优势在于集成了 BSTVC 的“全地图”单独建模框架，保证了拟合的局域空间回归系数具有直接可比性。空间非

平稳和时空非平稳是地理学第二定律的重要内涵。此外，无论是 BSTVC 还是 BSVC，在拟合非平稳性的时候，均考虑了基于地理学第一定律的时空自相关和空间自相关特征。

**时空方差分割指标 (Spatiotemporal Variance Partitioning Index, STVPI)** 基于 BSTVC/BSVC 建模结果，通过量化并比较不同时空异质影响因素的可解释百分比（时空贡献度/时空相对重要性）来明确关键驱动因素（[Song et al., 2022](#); [Wan et al., 2022](#)）。与当前主流的依赖绝对评价指标进行因子重要性排序的方法不同，STVPI 是一种相对评价指标，因此能够为地理时空归因提供重要的先验依据（[Wan et al., 2022](#)）。

## 1.2 BSTVC 方法优势

- **“全地图”单独建模框架：**该框架通过完整统一的贝叶斯层次建模机制，确保了局部时空回归系数的直接可比性，同时具备极强的拓展性，能够应对更复杂的实际应用挑战。正是由于该框架的设计，才使得计算相对时空贡献度（STVPI）成为可能。相比之下，在频率统计体系下，类似的分析通常采用“局部分开建模”的方式，即针对每个地图单元单独建模后再进行组合，可能引发一系列问题，例如不同小模型之间的可比性不足。
- **参数不确定性：**无论是目标变量的局域预测，还是时空回归系数的局域拟合，BSTVC 都能直接输出参数的不确定性评估结果，包括两种贝叶斯可信区间（50% 和 95%），以宽窄区间形式呈现。而在频率统计方法中，受限于统计机理，类似的分析无法评估不确定性。
- **缺失值友好：**即使目标变量  $Y$  存在较多时空缺失值，或解释变量  $X$  存在少量缺失值，也不影响时空非平稳效应的识别。而在频率统计方法中，类似的分析通常不支持缺失值。
- **更多空间权重矩阵的支持：**不仅支持基于距离的和  $k$  临近的空间权重矩阵，还支持 10 邻接矩阵来刻画数据中的空间自相关效应。而在频率统计方法中，类似的分析通常不支持 10 邻接矩阵。

## 1.3 R 包功能特点

- **面向多种目标变量：**支持三种主流目标变量类型：连续型（log-Gaussian）、二分类（logistic）和计数型（Poisson），满足不同分析场景的需求。
- **探测时空异质影响机制：**通过拟合时空回归系数，揭示解释变量（ $X$ ）与目标变量（ $Y$ ）之间的局域时空差异，深入分析“因地制宜、因时制宜”的规律，探索时空异质性带来的影响机制。
- **明确时空驱动因素：**在识别时空异质影响机制的基础上，通过计算时空可解释百分比，明确关键驱动因素，为地理时空归因提供有力证据。
- **提升时空预测精度：**考虑局域变量关系的时空异质性，显著提高模型拟合度和预测精度，用于时空缺失值填补、时空平滑和未来预测等。
- **贝叶斯模型评价：**提供贝叶斯模型的全面评估，包括模型拟合度（DIC、WAIC）、复杂度（pd）与预测精度（LS）等指标，帮助用户全面了解模型性能。
- **丰富的可视化输出：**同步提供多种时空可视化工具和代码，帮助用户直观理解模型结果，增强数据分析的可解释性，推动您的应用研究创新。

## 2 R 包安装

### 2.1 从 GitHub 上安装包

目前此包仅支持从 GitHub 上进行本地安装，安装方法如下：

```
# 使用 devtools 包安装
install.packages("devtools")           # 首先安装 devtools 包
devtools::install_github("songbi123/BSTVC") # 安装 BSTVC 包

# 使用 remotes 包安装
install.packages("remotes")           # 首先安装 remotes 包
remotes::install_github("songbi123/BSTVC") # 安装 BSTVC 包
```

### 2.2 安装依赖包

在 RStudio 中安装 BSTVC 包时，系统会提示您安装附带的其他 R 包。然而，由于依赖包 INLA 是一个较大的安装包，安装 BSTVC 包时可能会导致安装失败。为避免此问题，我们提供了独立安装 INLA 包的方法供用户参考。如果上一步安装 BSTVC 包失败，请在 INLA 包安装成功之后再行进行 BSTVC 包的安装。如果您在安装 BSTVC 包时已经安装好了 INLA 包，则可以跳过此步骤。

```
## 安装 INLA 包，更多信息访问 https://www.r-inla.org/download-install
options(timeout = 300) # 将超时时间延长到 5 分钟
install.packages("INLA", repos=c(getOption("repos"), INLA="https://inla.r-inla-download.org/R/stable"), dep=TRUE)

## 或使用旧版本的 INLA，您可以下载 INLA R 包的压缩包到本地再进行安装。
```

Note: 在手动安装 INLA 包时，请确保 INLA 包的版本与您的 R 版本兼容。如果您的 R 版本已更新至最新版，可能会出现没有与此版本 R 兼容的 INLA 包，从而导致安装失败。此外，在安装 BSTVC 包之前，请确保您的 R 版本为 4.4.1 及以上。

参考：开发 BSTVC R 包（V 25.1.28）时所用 R 版本为 4.4.1，所安装 INLA 包版本为 24.06.27。

### 2.3 加载 BSTVC 包

安装成功后直接加载即可使用。

```
library(BSTVC)
```

## 3 利用 BSTVC 函数构建局域时空回归模型

### 3.1 导入时空面板数据与检查

本教程以 BSTVC 包自带的示例数据为例进行数据导入，用户可以根据自己的数据进行相应操作。

需要加载两类数据文件：一是包含所有变量的表格文件，二是包含地理单元唯一值字段的 `sf` 格式 `shapefile` 文件。需要注意的是，如果数据包含多个时间截面，最终用于建模的表格数据应调整为时空面板格式（即长数据格式），以支持时空面板数据分析。

在 R 包中的 `data-raw/Data_Preproc.R` 文件中，公开展示了如何加载表格数据的源文件和地图数据文件（`shp` 格式），以及如何将表格数据转换为 BSTVC 模型可运行的时空面板数据格式。用户可以参考该文件中的示例代码，将相同的转换应用于自己的数据集。

示例数据集 `Florida_NAT` 展示了转换后的数据样式，用户可以使用 `data(Florida_NAT)` 加载示例数据进行参考。此外，关于将截面数据转换为时空面板数据的方法，用户可参考 `reshape2` 包中的 `melt` 函数、`tidyr` 包中的 `pivot_longer` 函数等，方法不止一种。

```
## 加载示例数据
# 包含所用的所有变量的表格文件（已是时空面板数据格式）
data(Florida_NAT)
# 包含地理单元唯一值字段的 sf 格式的 shapefile 文件
data(Florida_Map)

# 可用 class() 检查数据类型
class(Florida_NAT)
class(Florida_Map)

## 用户可参考以下方法自行导入自己的数据，但请注意数据类型
## 有关加载数据的详细步骤、转换数据的详细代码，已在 data-raw/Data_Preproc.R 中给出
# data <- read.csv("NAT.csv")
# data <- openxlsx::read.xlsx("NAT.xlsx")

# 加载 shapefile 数据，建议使用 sf 包中的 st_read 函数进行导入，
# 导入后 map 的类型应为 sf
# map <- sf::st_read("NAT.shp")

# class(data)
# class(map)
```

【备注】在 BSTVC 支持的时空面板数据中，变量 Y 可以包含缺失值，变量 X 也可容忍少量缺失值，缺失值请使用 NA 表示。

在用户导入自己的时空面板数据和地图数据后，需要检查两类数据中的字段类型。具体来说，需确保以下几点：

1. 数据框中表示年份的字段应为 `numeric` 类型。
2. 表示空间单元唯一标识符的字段应为正确的数据类型 (`numeric` 或 `character`)。
3. 地图数据中表示空间单元唯一标识符的字段类型应正确，并与数据框中的该字段名称和类型一致。

确保关键字段的类型正确且一致，能够避免在建模过程中出现错误。您可以使用 `str()` 函数检查字段类型，如下所示：

```
# 示例数据中，Year 字段代表这个时空面板数据的时间字段，FIPS 代表佛罗里达州每个区县的唯一标识符字段
# 且 Florida_Map 数据中必须存在与之相同的 FIPS 字段
str(Florida_NAT)
str(Florida_Map)

# 若需更改字段数据类型，可使用 as.numeric(), as.character 等函数进行转换
```

此外，还需要检查数据表格中关键字段（即表示每个空间单元的唯一值字段）的排序是否与 `map` 数据中关键字段的排序完全一致，不一致会造成最终结果不正确，但模型运行并不会报错的情况。例如，`Florida_Map` 中代表区县单元的 `FIPS` 字段的顺序为 12063, 12059, 12131, 12133, 12089，则 `Florida_NAT` 数据中的 `FIPS` 字段也必须为此排序。

针对此情况，可调用包中的 `data.check()` 函数进行检查。对于时空面板数据，`data.check()` 函数共包括 4 个参数，即：

- `data`: 导入的数据表格，必须包含代表时间字段与代表空间单元唯一标识符的字段
- `study_map`: 导入的 `sf` 格式的 `map` 数据，同样必须包含代表空间单元唯一标识符的字段
- `Time`: `data` 中表示时间的字段，输入字符串即可，即时间字段的列名
- `Space`: `data` 与 `study_map` 中表示空间单元唯一标识符的字段，输入字符串即可

```
# 在 R 中也可使用 ?data.check 来查看该函数的帮助文档
newdata <- data.check(data = Florida_NAT,
                      study_map = Florida_Map,
                      Time = "Year",    ## 输入字符串，不需要时间字段的向量数据
                      Space = "FIPS")

# 若空间单元顺序完全匹配，则函数结束运行时将会提示消息，且输出的 `newdata` 与函数中输入的 `data` 相同；
# 若顺序不一致，输出的 `newdata` 则是重新匹配好的新数据；
```

### 3.2 解释变量预处理

在进行回归建模之前，处理解释变量（`X` 变量）是一个至关重要的步骤。用户可以选择不对 `X` 变量进行任何预处理，或者对其进行对数（`log`）转换或标准化（`standardization`）处理。

我们强烈推荐对 `X` 变量执行标准化处理，原因如下：



- **标准化处理：**标准化后的 X 变量能够确保所有指标在同一尺度上进行比较，有助于衡量各个变量对模型的贡献大小。
- **提升计算效率：**标准化 X 变量还能够加速模型的计算过程。

【注意】针对连续型数据，由于模型内部使用了 log-Gaussian 先验，目标变量 Y 会自动进行对数转换，因此无需对 Y 进行 log 变换，直接输入原始数据即可。对于计数型和二分类型的 Y 变量，无需任何转换。

# 对示例数据中的解释变量进行标准化

```
Florida_NAT[c("DNL", "UE", "FP", "BLK", "GI")] <- scale(Florida_NAT[c("DNL", "UE", "FP", "BLK", "GI")])
```

# 在准备好所有数据之后，则可进行下一步的 BSTVC 函数建模

【备注】解释变量 X 的预处理和数据检查并无严格的先后顺序。

### 3.3 BSTVC 建模

BSTVC 包中提供的主要建模函数为 `BSTVC()` 函数，即贝叶斯时空变系数（Bayesian STVC）建模。该函数的参数主要有 8 个，具体参数描述如下：

- **formula:** 模型公式，基本格式为  $Y \sim \text{ST}(X1 + X2 + X3)$ ，用于指定建模中的目标变量与解释变量。在 BSTVC 函数中，所有解释变量 X 必须放在 `ST()` 符号内，表示这些解释变量同时具有空间和时间的非平稳性（目前仅支持这种情况）。
- **data:** 一个数据框，包含模型中所用的所有变量。若数据涉及多个时间截面，必须采用时空面板数据格式。每个时间截面下空间单元的顺序需要与 shapefile 文件中地理单元的顺序完全一致。
- **study\_map:** sf 格式数据，由 shapefile (*shp*) 格式的地图数据导入，其中包含每个地理单元的唯一值字段以及几何属性信息等内容。
- **Time:** 字符串，用于指定数据框 data 中的时间字段，只需提供时间字段的列名（字符串）即可。
- **Space:** 字符串，用于指定数据框 data 中代表空间单元的唯一值字段。需确保传入的字符串参数与 data 数据框中的列名完全一致，且 data 与 study\_map 中应同时具有名为此字符串的字段。
- **response\_type:** 字符串，指定建模数据的类型，目前可同时支持三种应用场景：“continuous”代表 Y 变量为连续型数据；“binary”代表 Y 变量为二分类型数据，即 0 表示不发生，1 表示发生；“count”代表 Y 变量为计数型变量。
- **threads:** 线程数，默认为 6，可自行设置。
- **spatial\_matrix:** 空间权重矩阵，默认为 10 邻接矩阵。用户可输入自行构建好的空间权重矩阵文件，且支持 k 临近、反距离、固定距离等多类型的空间权重矩阵；若用户未输入，则此函数将帮助计算 QUEEN 规则下的二分式（具有邻接关系的赋值为 1，否则赋值为 0）空间权重矩阵。

## ## BSTVC 函数建模 ——连续型

# 连续型时空数据建模，Y 变量为连续型数据，示例数据中使用 *HR*（每个区县的谋杀率）字段进行建模

# 指定时间字段为 "*Year*"，空间字段为 "*FIPS*"，设置 *response\_type* 参数为 "*continuous*"

# 默认 *spatial\_matrix* 参数为空，即 10 邻接矩阵

```
model_1 <- BSTVC(formula = HR~ST(DNL+UE+FP+BLK+GI),
                  data = Florida_NAT,
                  study_map = Florida_Map,
                  Time = "Year",
                  Space = "FIPS",
                  response_type = "continuous",
                  threads = 6,
                  spatial_matrix = NULL)
```

## ## BSTVC 函数建模 ——计数型

# Y 变量为计数型数据，示例数据中使用 *HC*（每个区县的谋杀案计数）字段进行建模

# 修改 Y 变量，设置 *response\_type* 参数为 "*count*"

```
model_2 <- BSTVC(formula = HC~ST(DNL+UE+FP+BLK+GI),
                  data = Florida_NAT,
                  study_map = Florida_Map,
                  Time = "Year",
                  Space = "FIPS",
                  response_type = "count",
                  threads = 6,
                  spatial_matrix = NULL)
```

## ## BSTVC 函数建模 ——二值型

# Y 变量为二值型数据，示例数据中使用 *HW*（有谋杀案发生则赋值为 1，否则为 0）字段进行建模

# 修改 Y 变量，设置 *response\_type* 参数为 "*binary*"

```
model_3 <- BSTVC(formula = HW~ST(DNL+UE+FP+BLK+GI),
                  data = Florida_NAT,
                  study_map = Florida_Map,
                  Time = "Year",
                  Space = "FIPS",
                  response_type = "binary",
                  threads = 6,
                  spatial_matrix = NULL)
```

### 3.4 模型结果

BSTVC 函数输出的结果共包括 6 个部分，具体输出部分描述如下：



| 输出结果                          | 描述  |
|-------------------------------|---|
| <b>model.evaluation</b>       | 贝叶斯模型的整体评价结果，其中包括 DIC、WAIC、LS 等常用的评估指标；   |
| <b>local.prediction</b>       | 目标变量 Y 的局部时空预测结果，以及每个预测值的宽 (95%)、窄 (50%) 贝叶斯可信区间，用于表达不确定性；  |
| <b>summary.random.effects</b> | 随机效应结果，其中包括每个解释变量的空间随机效应与时间随机效应等，及其宽 (95%)、窄 (50%) 贝叶斯可信区间；   |
| <b>time.coefficients</b>      | 时间回归系数 TCs（时间非平稳），宽数据格式，包括解释变量在每个时间切片的时间回归系数及其宽 (95%)、窄 (50%) 贝叶斯可信区间；  |
| <b>space.coefficients</b>     | 空间回归系数 SCs（空间非平稳），宽数据格式，包括解释变量在每个地图单元的空间回归系数及其宽 (95%)、窄 (50%) 贝叶斯可信区间；  |
| <b>STVPI</b>                  | 时空方差分割指标（STVPI）计算结果，可量化每个解释变量的时空贡献百分比，不仅能得到因子重要性的时空排序，还能评估相对重要性。有关此工具的更多信息，可参考文献 <a href="#">Wan et al., 2022</a> ； |

主函数 BSTVC 的输出结果中的每一部分都是一个数据框，用户可以直接提取并保存，或用于后续的结果可视化。以上面返回三个示例模型中的 `model_1` 为例，具体如下：

```
# 提取数据
# 局部单元预测结果（所有单元）
predict <- model_1$local.prediction # 此数据同样为时空面板格式
# 时间回归系数
time.coef <- model_1$time.coefficients
# 空间回归系数
space.coef <- model_1$space.coefficients
# 空间回归系数数据为截面数据格式，提取出后可直接保存为 csv 或 xlsx 格式，
# 并在 ArcGIS、ArcGIS Pro 等专业制图软件中进行系数分布地图的绘制，如下：
write.csv(space.coef, "空间回归系数.csv")
```

### 3.5 BSTVC 函数的输出及其可视化

在本文档中，我们将详细提供一系列的绘制模型输出结果的步骤和代码示例，以便您能够轻松地在 R 环境中创建各种图表。我们的目标是让您不仅能够理解模型的输出，还能够通过图表的形式，将这些复杂的数据以更直观、更吸引人的方式呈现出来。我们相信，通过本文档的帮助，您将能够更加自信地使用、更加熟练地上手我们的 R 包，并在您的数据分析工作中发挥更大的作用。

本文档在此部分展示的所有图表和可视化结果均基于示例数据构建的 `model_1` 模型的输出。以下绘图代码及图件供参考。

```
# 加载绘图 R 包
library(ggplot2)
library(dplyr)
library(ggthemes)
library(forestplot)
library(ggbeeswarm)
library(tidyr)
```

### 3.5.1 模型的贝叶斯评价

模型的贝叶斯评价包括模型名称（包含建模起始时间）、DIC、LS、WAIC、pd2 等指标。其中，DIC、WAIC 表示模型拟合度，eff（DIC）和 pd2（WAIC）表示模型复杂度，LS 表示预测精度。具体而言，DIC 和 WAIC 值越小表示模型拟合度越好，eff 和 pd2 值越小表示模型复杂度越低，LS 值越接近 0 表示预测精度越高。

```
# 从函数输出结果中提取贝叶斯评价结果。
model.eval <- model_1$model.evaluation
# 显示模型评价结果
model.eval
# 转置数据框进行指标展示
t(model.eval)
# 保存模型评估结果到本地
write.csv(model.eval, "results of model evaluation.csv")
```

```
ModelName      "BSTVC_Model: Start at 2025-01-04 16:45:01.163561,   end at 2025-01-04 16:45:42.286435."
DIC             "-1767.436"
eff            "77.66654"
LS             "-3.415557"
running_time   "0.6825578"
waic           "-1831.846"
pd2            "11.4044"
```

### 3.5.2 目标变量的时空预测

对于数据中目标变量 Y 的所有缺失值或非缺失值，局部预测结果中都会输出预测值，以及预测值的贝叶斯宽窄可信区间，用于直接评价不确定性。

```
# 提取局部预测数据框
predict <- model_1$local.prediction
# 查看输出字段
str(predict)
```

```
'data.frame': 268 obs. of 17 variables:
 $ Year      : num  1960 1960 1960 1960 1960 1960 1960 1960 1960 1960 ...
 $ FIPS      : num  12063 12059 12131 12133 12089 ...
 $ log(y)_mean : num  2.83 2.36 2.58 1.78 2.67 ...
 $ log(y)_sd   : num  0.000574 0.000626 0.000577 0.00056 0.000564 ...
 $ log(y)_0.025quant : num  2.83 2.36 2.58 1.78 2.67 ...
 $ log(y)_0.25quant : num  2.83 2.36 2.58 1.78 2.67 ...
 $ log(y)_0.5quant : num  2.83 2.36 2.58 1.78 2.67 ...
 $ log(y)_0.75quant : num  2.83 2.36 2.58 1.78 2.67 ...
 $ log(y)_0.975quant : num  2.83 2.36 2.58 1.78 2.67 ...
 $ y          : num  16.92 10.56 13.21 5.91 14.41 ...
 $ predict_y   : num  16.92 10.56 13.21 5.91 14.41 ...
 $ predict_0.025quant : num  16.9 10.6 13.2 5.9 14.4 ...
 $ predict_0.25quant : num  16.91 10.56 13.2 5.91 14.41 ...
 $ predict_0.5quant : num  16.92 10.56 13.21 5.91 14.41 ...
 $ predict_0.75quant : num  16.92 10.57 13.21 5.91 14.42 ...
 $ predict_0.975quant : num  16.94 10.58 13.22 5.92 14.43 ...
 $ smooth_y    : num  16.92 10.56 13.21 5.91 14.41 ...
```

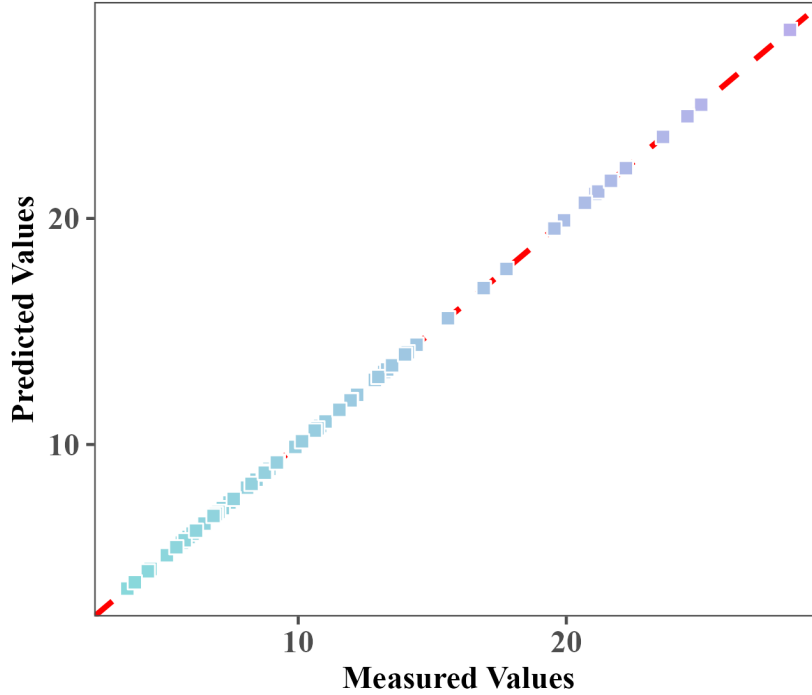
其中，对于变量 Y 所有类型（连续型、计数型与二值型）的局部预测结果，输出数据框中前两个字段是建模数据 `data` 中的时间字段与空间字段，`y` 字段代表目标变量 Y 的原始值，`fill_y` 字段代表对目标变量 Y 中缺失值进行预测填补后的完整字段，`predict_y` 代表为所有单元的模型预测值。

【注意】针对连续型的变量 Y，由于模型设置的是 log-Gaussian 先验分布，目标变量 Y 会自动进行对数变换，故我们对输出结果进行了再变换，并设置“predict\_”前缀。因此，(`predict_0.025quant`, `predict_0.975quant`) 即为预测值的宽可信区间，(`predict_0.25quant`, `predict_0.75quant`) 即为预测值的窄可信区间；

【注意】而对于其余两类 Y（二值型和计数型）的预测输出结果，目标变量 Y 没有做任何变换。因此输出的 (`y_0.025quant`, `y_0.975quant`) 即为预测值的宽可信区间，(`y_0.25quant`, `y_0.75quant`) 即为预测值的窄可信区间。

此外，基于局部预测的结果，可进行主流的预测精度（如散点图、R2、RMSE 等）的评价。下面以连续型建模结果为例，我们给出绘制散点图和计算 R2、RMSE 的代码：

```
## 基于 Y 的原始值与预测值，绘制散点图
(p1 <- ggplot(predict, aes(x = y, y = predict_y)) +
  # 绘制对角线
  geom_abline(slope = 1, intercept = 0, color = "red", size = 1, linetype = "dashed") +
  geom_point(aes(fill = y), size = 2.5, shape = 22, color = "white") +
  scale_fill_gradient(low = "#88d8db", high = "#b8aeeb") +
  labs(x = "Measured Values", y = "Predicted Values") +
  theme_few() +
  theme(
    text = element_text(family = "serif", face = "bold"),
    axis.text = element_text(family = "serif", size = 12, face = "bold"),
    legend.position = "none",
    axis.title = element_text(size = 12)
  )
)
```



```
## 基于 Y 的原始值与预测值，计算预测精度评估指标
# 提取没有缺失值的部分进行指标计算
predict <- predict %>% filter(!is.na(. $y))
# 手动计算评估指标：决定系数 ( $R^2$ )、均方根误差 (RMSE)
R2 <- 1 - ((sum((predict$predict_y - predict$y)^2)) / (sum((mean(predict$y) - predict$y)^2)))
RMSE <- sqrt(mean((predict$predict_y - predict$y)^2))
```

### 3.5.3 时空贡献度评价

时空方差分割指标 (Spatiotemporal Variance Partitioning Index, STVPI) 旨在量化并比较不同时空异质影响因素的可解释百分比，通过计算时空贡献度来明确关键驱动因素。基于 BSTVC 建模结果，STVPI 进一步将每个解释变量的总方差分解为时间和空间两个独立的组分，即时间非平稳随机效应和空间非平稳随机效应，进而揭示数据的时空变异来源。

此工具的优点有：

1. 不仅能识别解释因子的时空总体贡献，还能分别识别时间和空间维度的贡献；
2. 相比只能识别绝对贡献排序的主流方法（如随机森林、SHAP 等），STVPI 识别的是相对贡献（可解释百分比），为地理时空归为提供重要证据基础；
3. 直接评价其不确定性（贝叶斯宽窄可信区间）；

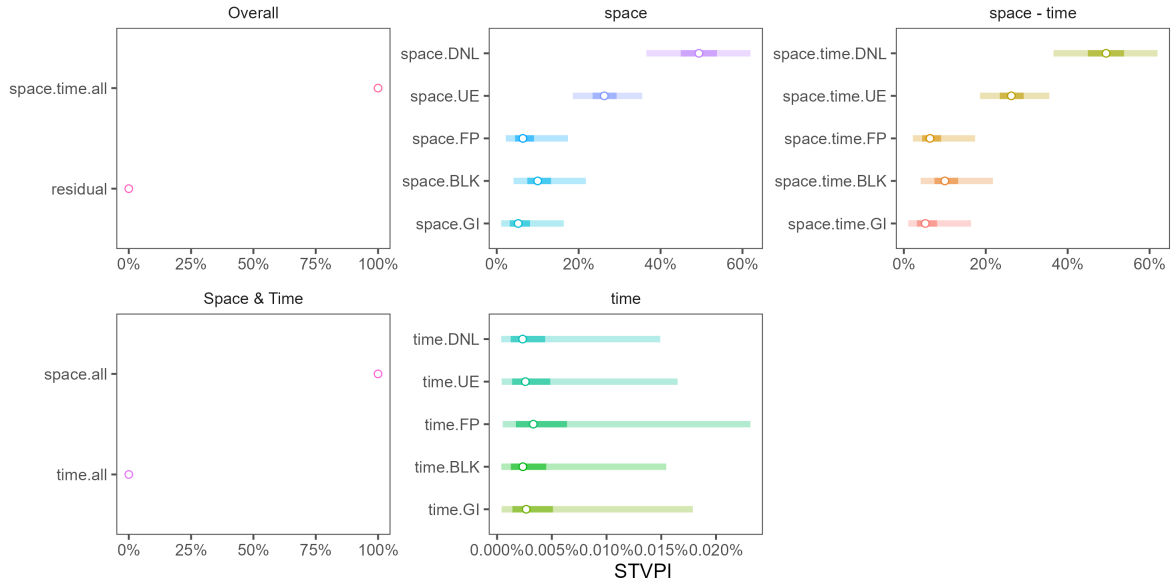
```
# 提取模型输出中的 STVPI 结果
STVPI.data <- model_1$STVPI

# 提取制图数据
```

```
## 此例为连续型结果的制图，若为计数型或二值型，需删除 "space.time.all","residual" 字段
random.effects <- c(
  "space.time.all", "residual", # 模型可解释度与残差
  "space.all", "time.all", # 时间 or 空间总贡献度
  "space.DNL", "space.UE", "space.FP", "space.BLK", "space.GI", # 每个 X 的空间贡献度
  "time.DNL", "time.UE", "time.FP", "time.BLK", "time.GI", # 每个 X 的时间贡献度
  "space.time.DNL", "space.time.UE", "space.time.FP", "space.time.BLK", "space.time.GI"
) # 每个 X 的时空贡献度
# 筛选数据并对数据进行分组，便于后续制图
plot.data <- as.data.frame(STVPI.data[STVPI.data$`random effects` %in% random.effects, ]) %>%
  mutate(Group = case_when(
    `random effects` %in% random.effects[1:2] ~ "Overall",
    `random effects` %in% random.effects[3:4] ~ "Space & Time",
    `random effects` %in% random.effects[5:9] ~ "space",
    `random effects` %in% random.effects[10:14] ~ "time",
    `random effects` %in% random.effects[15:19] ~ "space - time"
  )) %>%
  filter(Group != "NA")

# 逆转因子的展示顺序
plot.data$`random effects` <- factor(plot.data$`random effects`, levels = rev(random.effects))

# 绘制 STVPI 图
(p2 <- ggplot(plot.data, aes(x = STVPI_mean, y = `random effects`, group = Group)) +
  geom_errorbarh(aes(xmin = `STVPI_2.5%`, xmax = `STVPI_97.5%`,
    colour = `random effects`),
    height = 0, linewidth = 1.8, alpha = 0.3, linetype = "solid") +
  geom_errorbarh(aes(xmin = `STVPI_25%`, xmax = `STVPI_75%`,
    colour = `random effects`),
    height = 0, linewidth = 1.8, alpha = 0.6, linetype = "solid") +
  geom_point(aes(colour = `random effects`), shape = 21,
    fill = "white", size = 1.8, alpha = 1) +
  labs(title = "", x = "STVPI", y = "") +
  facet_wrap(~Group, scale = "free", ncol = 3) +
  theme_few() +
  scale_x_continuous(labels = scales::percent_format()) +
  theme(legend.position = "none"))
```



**【注意】** 泊松-BSTVC（计数型）和 logistic-BSTVC（二值型）模型目前未包括残差项，因此 STVPI 无法计算整体模型的可解释百分比。故在制图时，需删除 `random.effects` 变量中的“space.time.all”，“residual” 字段，以及 `plot.data` 变量中的“Overall” 字段。

#### 3.5.4 时间回归系数（时间非平稳）

BSTVC 模型不仅可以计算时间回归系数来刻画变量关系的时间异质性，还可以估算出每个回归系数值的贝叶斯宽（95%）、窄（50%）可信区间，用于直接评价结果的不确定性。

```
# 提取模型输出中的时间回归系数，添加系数字段与可信区间字段
Time.Coeff <- model_1$time.coefficients %>%
  mutate(
    Coefficients = round(mean, 2),
    `95%CI` = paste0("(", round(`0.025quant`, 2), " - ", round(`0.975quant`, 2), ")")
  )

# 将时间字段转换为年份（可做可不做）
Time.Coeff <- Time.Coeff %>% mutate(Year = case_when(
  time_index == 1 ~ 1960,
  time_index == 2 ~ 1970,
  time_index == 3 ~ 1980,
  time_index == 4 ~ 1990
))

# 添加表头
new.Time.Coeff <- rbind.data.frame(colnames(Time.Coeff), Time.Coeff)
new.Time.Coeff[1, c("mean", "0.025quant", "0.975quant")] <- NA
```

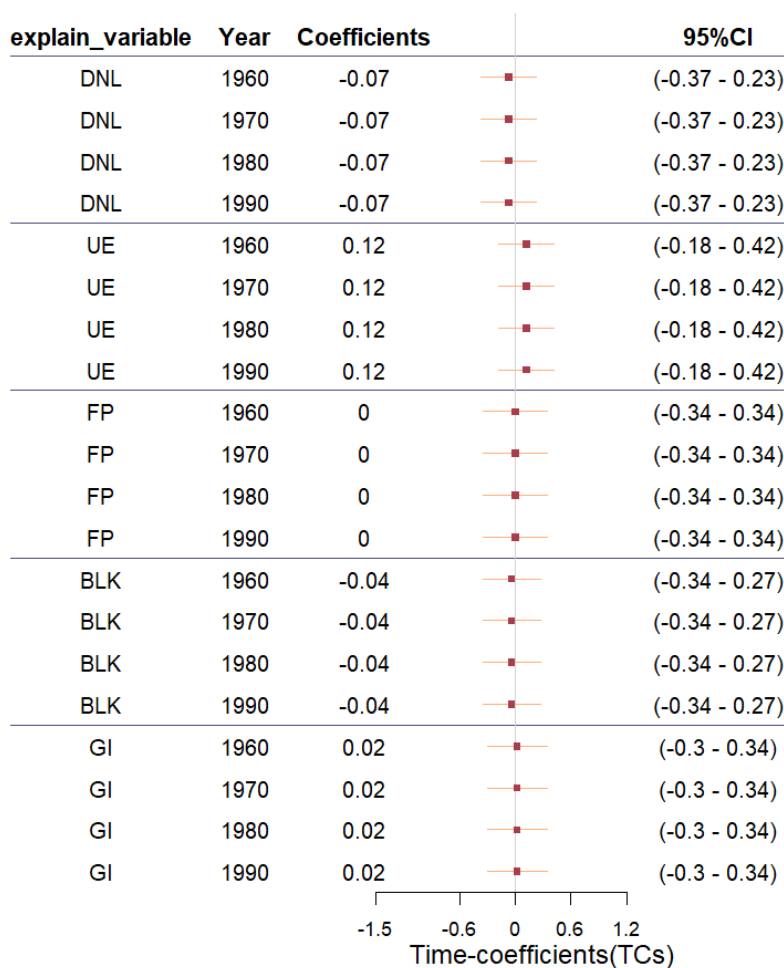
## 时间回归系数的森林图展示形式

```
(p3 <- forestplot(new.Time.Coeff[, c(9, 12, 10, 11)],
  mean = new.Time.Coeff$Coefficients,
  lower = new.Time.Coeff$`0.025quant`,
  upper = new.Time.Coeff$`0.975quant`,
  graph.pos = 4,
  xlab = "Time-coefficients(TCs)",
  is.summary = c(T, rep(FALSE, nrow(Time.Coeff))),
  align = "c",
  graphwidth = unit(4, "cm"),
  xticks = (c(-1.5, -0.6, 0, 0.6, 1.2)),
  col = fpColors(lines = "#ffb07c", box = "#a83e4c"),
  hrzl_lines = list(
    "2" = gpar(col = "#4e5180"), "6" = gpar(col = "#4e5180"), "10" = gpar(col = "#4e5180"),
    "14" = gpar(col = "#4e5180"), "18" = gpar(col = "#4e5180")
  ),
  txt_gp=fpTxtGp(label=gpar(cex=1.25),
    ticks=gpar(cex=1.1),
    xlab=gpar(cex = 1.5),
    title=gpar(cex = 1.2)),
  lineheight = unit(1, "cm"),
  boxsize = 0.15,
  ref_line = 0, # 添加参考线
))
```

## 时间回归系数的时间序列展示形式（第二种制图，供参考）

```
# (p3 <- ggplot(data=Time.Coeff,aes(x=Year,y= Coefficients ,fill=explain_variable))+
#   geom_ribbon(aes(ymin=`0.025quant`,ymax=`0.975quant`,fill=explain_variable),alpha=0.1)+
#   geom_ribbon(aes(ymin=`0.25quant`,ymax=`0.75quant`,fill=explain_variable),alpha=0.5)+
#   geom_line(aes(x=Year,y= Coefficients,colour=explain_variable),alpha=0.8)+
#   facet_wrap(~explain_variable,scale = "free_y")+
#   theme_few()+
#   labs(title=NULL, x="Time dimension", y="Time-coefficients (TCs)")
# )
```





### 3.5.5 空间回归系数（空间非平稳）

与时间回归系数类似，BSTVC 模型不仅可以计算空间回归系数表示空间异质的变量关系，还可以估算其不确定性，即贝叶斯宽（95%）、窄（50%）可信区间。

**3.5.5.1 绘制空间回归系数地图** 提取出模型输出中的空间回归系数数据框后，建议在 ArcGIS 或 ArcGIS Pro 等专业地图制图软件中进行绘制。此处基于 R 语言，简单绘制出基础的空间回归系数地图展示结果，仅供样本制图参考。

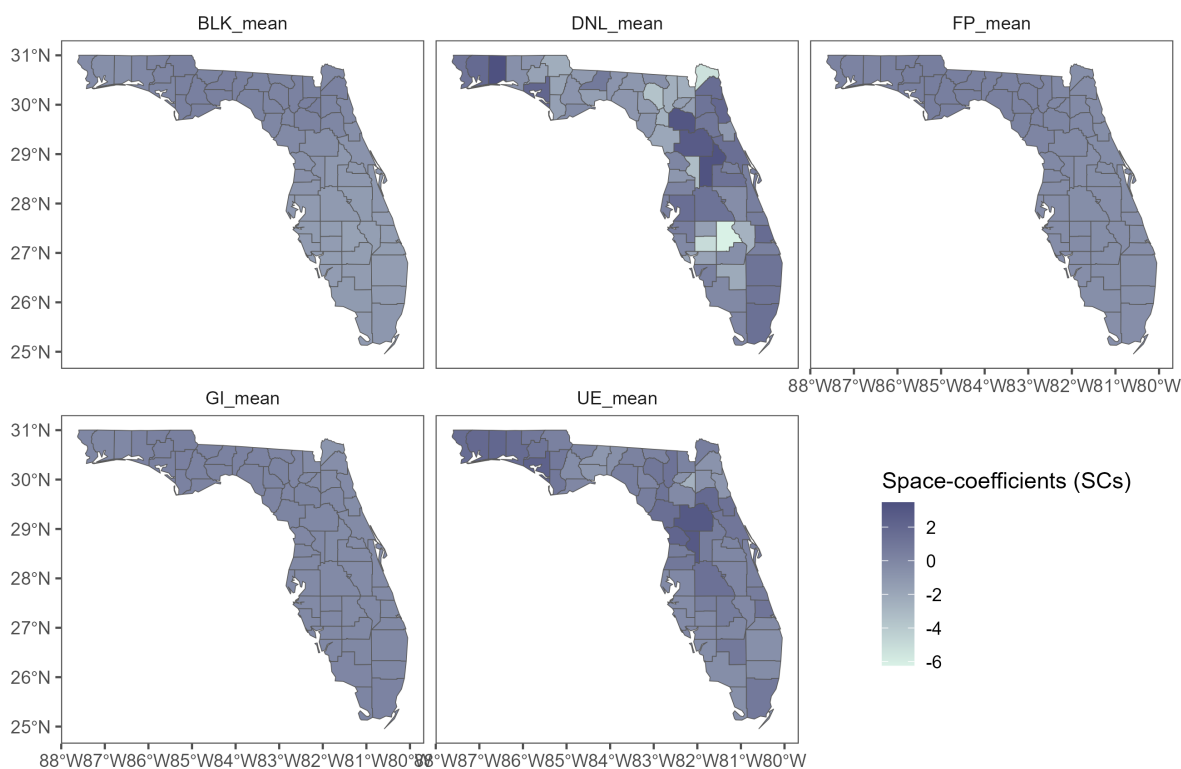
```
# 提取模型输出中的空间回归系数
Space.Coeff <- model_1$space.coefficients

# 组合地图中的几何属性与每个地理单元的空间回归系数
Space.Coeff.merge <- merge(Florida_Map, Space.Coeff, by = "FIPS", all.x = T)

# 将截面数据转换为面板数据格式
Space.coef.panel <- Space.Coeff.merge %>%
  pivot_longer(
```

```
cols = paste0(c("DNL", "UE", "FP", "BLK", "GI"), "_mean"),
names_to = "variable",
values_to = "SCs"
) %>%
select(c(1:9, 45:47))

# 绘制地图
(p4 <- ggplot() +
  geom_sf(data = Space.coef.panel, aes(group = variable, fill = SCs)) +
  facet_wrap(~variable, shrink = F, drop = F) +
  scale_fill_gradient(low = "#d9f1e6", high = "#4e5180", name = "Space-coefficients (SCs)") +
  theme_few() +
  theme(legend.position = c(.85, .25))
)
```



**3.5.5.2 单个空间单元的因子影响效应制图** 仿照 SHAP 制图范式，本图展示了单个空间单元的因子影响效应图。该图为两类图形的组合：

- 横向柱形图：每个横向柱形代表了某个空间单元的不同解释变量  $X$  的空间回归系数；
- 线段图：叠加在柱形图上的线段表示了这些回归系数的宽（95%）和窄（50%）贝叶斯可信区间。

这种可视化方法有助直观地理解不同空间单元对目标变量影响的空间异质性，以及估计系数的不确

定性范围。以下是绘制单个空间单元的空间回归系数的 R 代码，批量绘制多个空间单元的因子影响效应图的 R 代码可参考此小节的后部分。

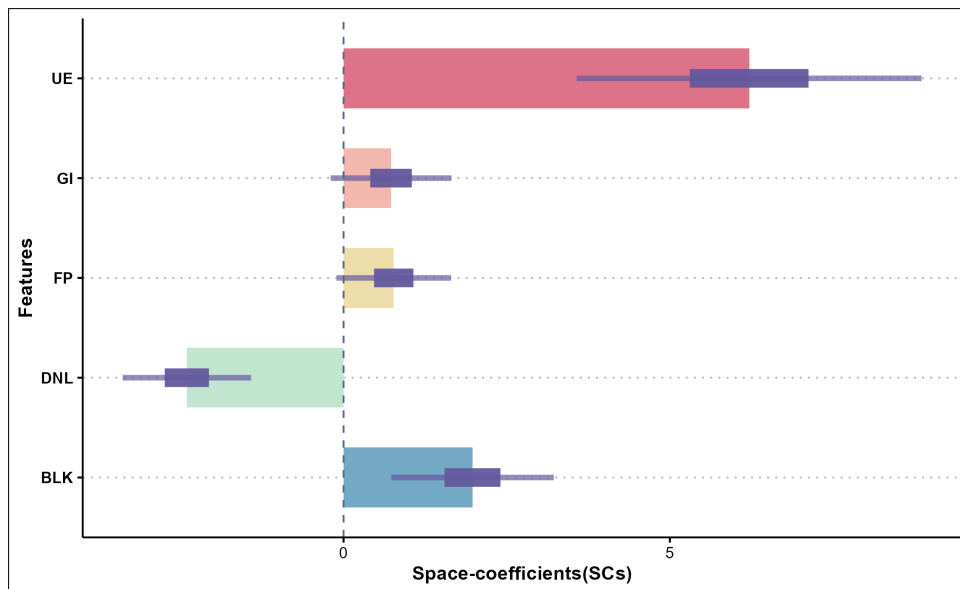
```
# 提取 X 名字
name.X <- gsub("_mean", "", names(Space.Coeff[, c(2, 10, 18, 26, 34)]))

# 手动处理数据 # 转换为制图数据（时空面板数据格式）
SC <- list()
for (name in c("mean", "0.025quant", "0.25quant", "0.75quant", "0.975quant")) {
  SC[[name]] <- Space.Coeff %>%
    pivot_longer(cols = paste0(c("DNL", "UE", "FP", "BLK", "GI"), "_", name),
                  names_to = "feature", values_to = name)
}

# 组合制图数据
impact_data <- cbind.data.frame(
  feature = rep(name.X, times = nrow(Space.Coeff)), # 变量
  SC$mean$FIPS,
  SC$mean$mean, SC$`0.025quant`$`0.025quant`, SC$`0.25quant`$`0.25quant`,
  SC$`0.75quant`$`0.75quant`, SC$`0.975quant`$`0.975quant`
) %>%
  `colnames<-`(c("feature", "FIPS", "mean", "0.025quant", "0.25quant",
                 "0.75quant", "0.975quant"))

## 绘制某个空间单元的因子影响效应
# 提取唯一值字段为 "12133" 的空间回归系数
single_data <- impact_data %>% filter(FIPS == 12133)
# 绘图
(p5 <- ggplot(data = single_data)+
  geom_bar(
    aes(x = mean, y = feature, fill = feature),
    stat = "identity", width = 0.6)+
  geom_segment(aes(x = `0.025quant`, y = feature,
                  xend = `0.975quant`, yend = feature),
               color="#63599d",size = 1.5,alpha=.7)+
  geom_segment(aes(x = `0.25quant`, y = feature,
                  xend = `0.75quant`, yend = feature),
               color="#63599d",size = 5,alpha=.95)+
  # 添加水平参考线
  geom_vline(xintercept = 0, linetype = "dashed", size = 0.5, color = "#4e6886") +
  scale_fill_manual(values = c("#74a9c5", "#c2e5cf", "#edddab", "#f2b8ae", "#dd7389")) +
  scale_color_manual(values = c("#74a9c5", "#c2e5cf", "#edddab", "#f2b8ae", "#dd7389")) +
  theme_clean()+
```

```
labs(x="Space-coefficients(SCs)",y="Features")+
theme(
  axis.text.y = element_text(size = 9, face = "bold"),
  axis.title.x = element_text(size = 11, face = "bold"),
  axis.title.y = element_text(size = 11, face = "bold"),
  legend.position = "none")
)
```



## 绘制多个空间单元的因子影响效应

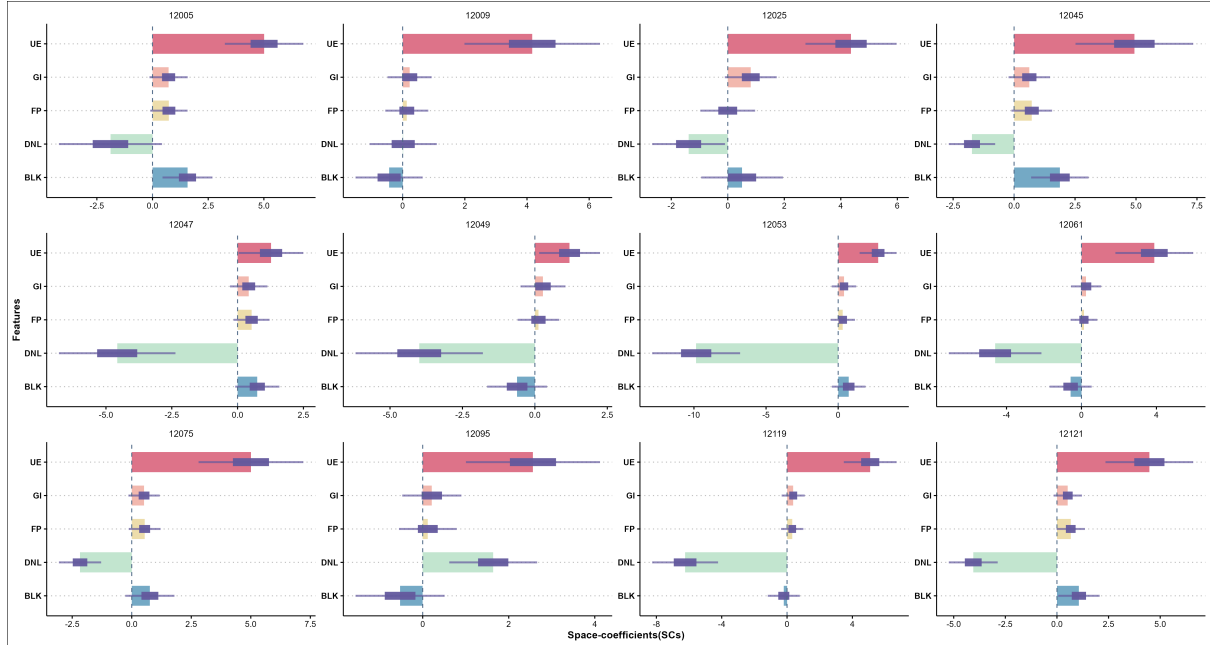
# 随机提取 12 个空间单元的空间回归系数

```
multi_data <- impact_data[impact_data$FIPS %in% sample(unique(impact_data$FIPS),12),]
```

# 绘图

```
(p6 <- ggplot(data = multi_data,
  aes(x = mean, y = feature, fill = feature))+
  geom_bar(stat = "identity", width = 0.6)+
  geom_segment(aes(x = `0.025quant`, y = feature,
    xend = `0.975quant`, yend = feature),
    color="#63599d",size = 1,alpha=.7)+
  geom_segment(aes(x = `0.25quant`, y = feature,
    xend = `0.75quant`, yend = feature),
    color="#63599d",size = 4,alpha=.95)+
  # 添加水平参考线
  geom_vline(xintercept = 0, linetype = "dashed", size = 0.5, color = "#4e6886") +
  scale_fill_manual(values = c("#74a9c5", "#c2e5cf", "#edddab", "#f2b8ae", "#dd7389")) +
  scale_color_manual(values = c("#74a9c5", "#c2e5cf", "#edddab", "#f2b8ae", "#dd7389")) +
  facet_wrap(~FIPS, scale="free",ncol=4)+
  theme_clean()+
  labs(x="Space-coefficients(SCs)",y="Features"))
```

```
theme(
  axis.text.y = element_text(size = 9, face = "bold"),
  axis.title.x = element_text(size = 11, face = "bold"),
  axis.title.y = element_text(size = 11, face = "bold"),
  legend.position = "none")
)
```



**3.5.5.3 空间回归系数的蜂巢-条状组合图** 仿照 SHAP 制图范式，本图展示了模型中各个变量的空间回归系数分布情况。每一行代表一个不同的解释变量，而每个点则对应于该变量在每个空间单元上的空间回归系数：

- 点的分布：图中的散点表示各个空间单元的空间回归系数，这些点的分布揭示了变量在空间上的影响程度和方向的变异性。
- 平均值柱形图：每个变量下方的柱形图显示了该变量所有空间回归系数的平均值（即代表全局空间回归系数），柱形图的颜色深浅表示平均系数的影响方向，从而提供了对变量整体影响的直观理解。

# 手动创建直方图数据

```
bar_data <- data.frame(
  feature = rep(name.X, each = nrow(Space.Coeff)), # 变量
  coef_value = c(Space.Coeff$DNL_mean, Space.Coeff$UE_mean, Space.Coeff$FP_mean,
                 Space.Coeff$BLK_mean, Space.Coeff$GI_mean) # 局部空间回归系数
) %>%
group_by(feature) %>%
summarise(
  mean = mean(coef_value),
```

```

    abs_mean = mean(abs(coef_value))
  )

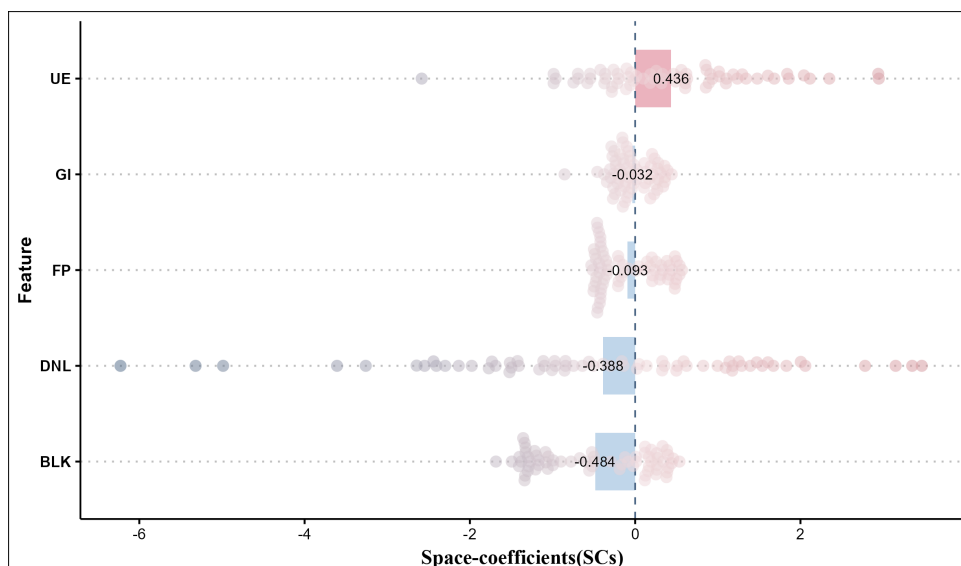
# 手动创建蜂窝图数据
beeswarm_data <- data.frame(
  feature = rep(name.X, each = nrow(Space.Coeff)), # 变量
  coef_value = c(Space.Coeff$DNL_mean, Space.Coeff$UE_mean, Space.Coeff$FP_mean,
    Space.Coeff$BLK_mean, Space.Coeff$GI_mean) # 局部空间回归系数
)

# 绘制组合图
(p7 <- ggplot() +
  ## 绘制平均的空间回归系数（全局空间回归系数）
  geom_bar(
    stat = "identity", data = bar_data,
    aes(x = mean, y = feature, fill = mean < 0), width = 0.6
  ) +
  scale_fill_manual(values = c("#EBB3BE", "#COD6EA")) +
  # 添加水平参考线
  geom_vline(xintercept = 0, linetype = "dashed", size = 0.5, color = "#4e6886") +

  ## 绘制局部空间回归系数分布图
  geom_beeswarm(
    data = beeswarm_data,
    aes(x = coef_value, y = feature, colour = coef_value), alpha = 0.5,
    cex = 1, shape = 16, size = 3, priority = "descending"
  ) +
  scale_colour_gradient2(low = "#4e6886", mid = "#eed8db", high = "#bb616d",
    midpoint = 0, name = NULL) + # 渐变颜色
  geom_text(
    data = bar_data, aes(x = mean, y = feature, label = round(mean, 3)),
    # position = position_stack(vjust = 30, reverse = T),
    size = 3,
    color = "black"
  ) + # 在柱形上添加数值
  theme_clean() +
  theme(
    plot.title = element_text(hjust = 0.5, color = "purple"),
    axis.text.y = element_text(size = 9, face = "bold"),
    axis.title.x = element_text(size = 11, face = "bold"),
    axis.title.y = element_text(size = 11, face = "bold"),
    legend.position = "none"
  )
)

```

```
) +
labs(
  title = NULL,
  x = "Space-coefficients(SCs)",
  y = "Feature"
))
```



## 4 利用 BSVC 函数构建局域空间回归模型

与贝叶斯时空变系数 (BSTVC) 模型相比, 贝叶斯空间变系数 (BSVC) 模型的主要区别在于, BSVC 模型不考虑时间维度上的非平稳性。

在这个案例中, 我们将额外展示如何构建自定义的空间权重矩阵, 以定义空间自相关特征。默认的空间权重矩阵是基于面状空间单元的 10 邻接矩阵, 用户无需自定义 (如本文档第 3 部分所示)。

### 4.1 导入空间截面数据与检查

由于 BSVC 模型主要适用于关注空间效应而无需考虑时间变化的场景, 因此不需要导入多个时间节点 (如:  $n$  年) 的数据, 只需提供一个时间切片的数据 (格式与截面数据相同)。在此示例中, 我们提取了示例数据中的最后一年数据作为 BSVC 模型的建模输入。对于用户自己的数据, 可使用 BSTVC 函数中相同的数据导入方法。

```
data <- Florida_NAT[Florida_NAT$Year == 1990,]
```

**【备注】** 在 BSVC 模型中, 变量 Y 可以包含缺失值, 变量 X 也可容忍少量缺失值, 缺失值请使用 NA 表示。

在用户导入自己的数据 (一个时间切片的时空面板数据或截面数据) 与地图数据后, 需要检查两类数据中的字段类型。具体来说, 需确认以下几点:



1. 数据框中表示年份的字段是否为 `numeric` 类型;
2. 表示空间单元唯一标识符的字段是否为正确的数据类型 (`numeric` 或 `character`);
3. 地图数据中表示空间单元唯一标识符的字段类型是否正确, 并且与数据框中的该字段名称和类型一致。

确保关键字段的类型符合要求, 并且两类数据中相同字段的类型一致, 可以有效避免建模过程中的错误。可以使用 `str()` 函数来检查字段类型, 如下所示:

```
# 示例数据中, Year 字段代表这个时空面板数据的时间字段, FIPS 代表佛罗里达州每个区县的唯一标识符字段,
# 且 Florida_Map 数据中必须存在与之相同的 FIPS 字段
str(data)
str(Florida_Map)

# 若需更改字段数据类型, 可使用 as.numeric(), as.character 等函数进行转换
```

对于 BSVC 建模的数据, 仍需确保数据表格中关键字段 (即表示每个空间单元唯一值的字段) 的排序与地图数据中的关键字段排序完全一致。此外, 即使数据没有时间维度, 也可以使用 `data.check` 函数进行空间单元顺序的匹配检查。所需的参数包括 `data`、`study_map`、`Time` 与 `Space`, 示例如下所示:

请注意, 即使输入的是空间数据, `Time` 参数 (数据中表示时间的字段) 仍然是必需的。

```
newdata <- data.check(data = data,
                      study_map = Florida_Map,
                      Time = "Year", # 输入字符串, 不需要时间字段的向量数据
                      Space = "FIPS")

# 若自己的数据中并无时间字段, 可自行添加一个, 例:
# data["year"] <- rep(1, each=nrow(data)) # 添加一个 值全为 1 的 year 字段
```

## 4.2 解释变量预处理

在进行回归建模之前, 处理解释变量 (X 变量) 是一个至关重要的步骤。用户可以选择不对 X 变量进行任何预处理, 或者对其进行对数 (`log`) 转换或标准化 (`standardization`) 处理。我们强烈推荐对 X 变量执行标准化处理, 优点在 BSTVC 案例部分已经阐述。由于示例数据已对 X 变量执行了标准化处理, 因此此处不再进行演示。

【备注】针对连续型数据, 由于模型内部使用了 `log-Gaussian` 先验, 目标变量 Y 会自动进行对数转换, 因此无需对 Y 进行 `log` 变换, 直接输入原始数据即可。对于计数型和二分类型的 Y 变量, 无需任何转换。

### 4.3 BSVC 建模（自定义空间权重矩阵）

执行 BSVC 建模的函数为 `BSVC()`，该函数参数的设置与 `BSTVC()` 函数相差无几，不同的地方主要体现在 `formula`、`data` 参数上，且此函数不需要 `Time` 参数。具体参数描述如下：

- **formula:** 模型公式，基本格式为  $Y \sim S(X_1 + X_2 + X_3)$ ，用于指定建模中的目标变量与解释变量。其中，BSVC 函数中的解释变量  $X$  需全部放于  $S()$  符号中，表示解释变量  $X$  存在空间非平稳性。
- **data:** 一个数据框，包含模型中所用的所有变量。该数据框应仅包含空间维度上变化的信息，即它应该代表时空面板数据集中的某一个时间切片，例如特定年份或特定月份的数据。`data` 中每年空间单元的顺序需与 `shp` 文件中每个地理单元的顺序完全一致。
- **study\_map:** `sf` 格式数据，由 `shapefile (shp)` 格式的地图数据导入，其中包含每个地理单元的唯一值字段以及几何属性信息等内容。
- **Space:** 字符串，用于指定数据框 `data` 中代表空间单元的唯一值字段。需确保传入的字符串参数与 `data` 数据框中的列名完全一致，且 `data` 与 `study_map` 中应同时具有名为此字符串的字段。
- **response\_type:** 字符串，指定建模数据的类型，目前可同时支持三种应用场景：“continuous”代表  $Y$  变量为连续型数据；“binary”代表  $Y$  变量为二分类型数据，即 0 表示不发生，1 表示发生；“count”代表  $Y$  变量为计数型变量。
- **threads:** 线程数，默认为 6，可自行设置。
- **spatial\_matrix:** 空间权重矩阵，默认为 10 邻接矩阵。用户可输入自行构建好的空间权重矩阵文件，且支持  $k$  临近、反距离、固定距离等多类型的空间权重矩阵；若用户未输入，则此函数将帮助计算 QUEEN 规则下的二分式（具有邻接关系的赋值为 1，否则赋值为 0）空间权重矩阵。

## BSVC 函数建模 ——连续型

# 连续型空间数据建模， $Y$  变量为连续型数据，示例数据中使用 `HR`（每个区县的谋杀率）字段中的  
# 最后一年数据进行建模

```
model_type1 <- BSVC(formula = HR~S(DNL+UE+FP+BLK+GI),
                     data = newdata,
                     study_map = Florida_Map,
                     Space = "FIPS",
                     response_type = "continuous",
                     threads = 6,
                     spatial_matrix = NULL)
```

## BSTVC 函数建模 ——计数型

#  $Y$  变量为计数型数据，示例数据中使用 `HC`（每个区县的谋杀案计数）字段进行建模  
# 修改  $Y$  变量，设置 `response_type` 参数为 "count"

```
model_type2 <- BSVC(formula = HC~S(DNL+UE+FP+BLK+GI),
                     data = newdata,
                     study_map = Florida_Map,
                     Space = "FIPS",
```

```

        response_type = "count",
        threads = 6,
        spatial_matrix = NULL)

## BSTVC 函数建模 —— 二值型
# Y 变量为二值型数据，示例数据中使用 HW（有谋杀案发生则赋值为 1，否则为 0）字段进行建模
# 修改 Y 变量，设置 response_type 参数为 "binary"
model_type3 <- BSVC(formula = HW~S(DNL+UE+FP+BLK+GI),
                    data = newdata,
                    study_map = Florida_Map,
                    Space = "FIPS",
                    response_type = "binary",
                    threads = 6,
                    spatial_matrix = NULL)

## spatial_matrix 参数设置的 NULL，即按照默认的 QUEEN 规则计算空间权重矩阵，若有需要，
## 可输入自定义的空间权重矩阵

# 如下例所示，在 R 中创建自定义的点要素的空间权重矩阵并进行建模
library(spdep)
library(sf)

# 提取 Map 的质心作为点要素对象：
nc.point <- st_centroid(st_geometry(Florida_Map))
# 使用函数 knearneigh 和 knn2nb 创建 k 最近邻空间邻接矩阵
nb_knn <- knn2nb(knearneigh(nc.point, k = 10))

## 计算 --k 最近邻-- 的空间权重矩阵
# 这里使用二元权重
weight_knn <- nb2mat(nb_knn, style = "B") ## 此为 k 最近邻空间权重矩阵

## 计算 --反距离-- 的空间权重矩阵
# 计算点之间的最大距离
linked <- max(unlist(nbdists(nb_knn, nc.point)))
# 使用 dnearneigh 函数生成基于距离的邻域
# 假设我们选择距离范围为 0 到最大距离
nb_dist <- dnearneigh(nc.point, 0, linked)

# 计算距离衰减式的空间权重矩阵

```

```
weight_list_dist <- nb2listwdist(nb_dist, x=nc.point,
                                type = "idw", style = "W",
                                zero.policy = T,alpha = 1)
# 使用 listw2mat 函数将前者 list 类型的空间权重矩阵转换为矩阵类型
weight_dist <- listw2mat(weight_list_dist)    ### 此为反距离空间权重矩阵

# 基于自定义的反距离空间权重矩阵重新进行 BSVC 建模
model_type1_new <- BSVC(formula = HR~S(DNL+UE+FP+BLK+GI),
                        data = newdata,
                        study_map = nc.point,
                        Space = "FIPS",
                        response_type = "continuous",
                        threads = 6,
                        spatial_matrix = weight_dist)
```

#### 4.4 模型结果

对比 BSTVC 函数，BSVC 函数除了没有 `time.coefficients` 的结果外，其余 5 个部分即是该函数的所有结果。具体输出部分描述如下：

| 输出结果                          | 描述  |
|-------------------------------|---|
| <b>model.evaluation</b>       | 贝叶斯模型的全局评价结果，其中包括 DIC、WAIC、LS 等常用的评估指标；   |
| <b>local.prediction</b>       | 目标变量 Y 的局部预测结果，以及每个预测值的宽 (95%)、窄 (50%) 贝叶斯可信区间，用于表达不确定性；  |
| <b>summary.random.effects</b> | 随机效应结果，其中包括每个解释变量的空间随机效应，以及宽 (95%)、窄 (50%) 贝叶斯可信区间；   |
| <b>space.coefficients</b>     | 空间回归系数 SCs，宽数据格式，包括每个地图单元的空间回归系数及其宽 (95%)、窄 (50%) 贝叶斯可信区间，用于刻画空间非平稳；  |
| <b>STVPI</b>                  | 时空方差分割指标 (STVPI) 计算结果，可量化每个解释变量的时空贡献百分比。这里将会输出每个解释变量 X 的空间贡献百分比。有关此工具的更多信息，可参考文献 <a href="#">Wan et al., 2022</a> ； |

#### 4.5 BSVC 函数的输出及其可视化

本文档在此部分展示的所有图表和可视化结果均示例数据构建的 `model_type1` 模型的输出。以下绘图代码及图件供参考。

```
# 加载绘图 R 包
library(ggplot2)
library(dplyr)
library(ggthemes)
library(ggbeeswarm)
library(tidyr)
```

#### 4.5.1 模型的贝叶斯评价

```
# 提取模型中数据，此为所有评估结果
model.eval.BSVC <- model_type1$model.evaluation
# 转置数据框进行指标展示
t(model.eval.BSVC)
# 保存模型评估结果
write.csv(model.eval.BSVC, "results of model evaluation.csv")

ModelName      "BSVC_Model: Start at 2025-01-06 15:52:30.485916,   end at 2025-01-06 15:52:33.393795."
DIC             "603.6244"
eff            "13.11141"
LS             "4.50647"
running_time   "0.0401963"
waic           "602.2904"
pd2            "10.01077"
```

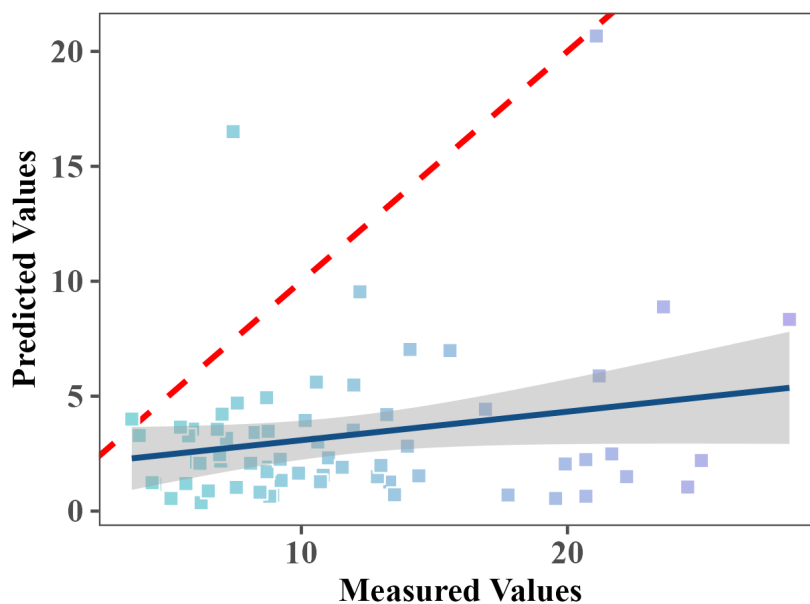
#### 4.5.2 目标变量的空间预测

```
# 提取局部预测数据框
predict.BSVC <- model_type1$local.prediction
# 查看输出字段
str(predict.BSVC)

'data.frame':   67 obs. of  16 variables:
 $ FIPS          : num  12063 12059 12131 12133 12089 ...
 $ log(y)_mean   : num  1.489 1.725 1.435 1.272 0.429 ...
 $ log(y)_sd     : num  0.783 1.148 0.852 0.68 0.628 ...
 $ log(y)_0.025quant : num  -0.0466 -0.5247 -0.2358 -0.0606 -0.8024 ...
 $ log(y)_0.25quant : num  0.9603 0.95071 0.85989 0.81336 0.00513 ...
 $ log(y)_0.5quant : num  1.489 1.725 1.435 1.272 0.429 ...
 $ log(y)_0.75quant : num  2.017 2.499 2.01 1.73 0.853 ...
 $ log(y)_0.975quant : num  3.02 3.97 3.11 2.6 1.66 ...
 $ y            : num  16.92 10.56 13.21 5.91 14.41 ...
 $ predict_y     : num  4.43 5.61 4.2 3.57 1.54 ...
 $ predict_0.025quant : num  0.955 0.592 0.79 0.941 0.448 ...
 $ predict_0.25quant : num  2.61 2.59 2.36 2.26 1.01 ...
 $ predict_0.5quant : num  4.43 5.61 4.2 3.57 1.54 ...
 $ predict_0.75quant : num  7.51 12.17 7.46 5.64 2.35 ...
 $ predict_0.975quant : num  20.57 53.22 22.32 13.52 5.26 ...
 $ smooth_y      : num  4.43 5.61 4.2 3.57 1.54 ...
```

基于局部预测的结果，可进行主流预测精度的常用评价，以连续型建模结果为例：

```
## 基于 Y 的原始值与预测值，绘制散点图
(p8 <- ggplot(predict.BSVC, aes(x = y, y = predict_y)) +
  # 绘制对角线
  geom_abline(slope = 1, intercept = 0, color = "red", size = 1, linetype = "dashed") +
  geom_point(aes(fill = y), size = 2.5, shape = 22, color = "white") +
  geom_smooth(method = "lm", se = T, color = "#134F85") +
  scale_fill_gradient(low = "#88d8db", high = "#b8aeeb") +
  labs(x = "Measured Values", y = "Predicted Values") +
  theme_few() +
  theme(
    text = element_text(family = "serif", face = "bold"),
    axis.text = element_text(family = "serif", size = 12, face = "bold"),
    legend.position = "none",
    axis.title = element_text(size = 12)
  )
)
```



```
## 基于 Y 的原始值与预测值，计算预测精度评估指标
# 提取没有缺失值的部分进行指标计算
predict.BSVC <- predict.BSVC %>% filter(!is.na(. $y))

# 手动计算评估指标：决定系数 ( $R^2$ )、均方根误差 (RMSE)
R2 <- 1 - ((sum((predict.BSVC$predict_y - predict.BSVC$y)^2)) /
  (sum((mean(predict.BSVC$y) - predict.BSVC$y)^2)))
RMSE <- sqrt(mean((predict.BSVC$predict_y - predict.BSVC$y)^2))
```

### 4.5.3 空间贡献度评价

以连续性 Y 变量为例，得到每个解释因子 X 的空间贡献百分比。

```
# 提取模型输出中的 STVPI 结果
STVPI.data.BSVC <- model_type1$STVPI

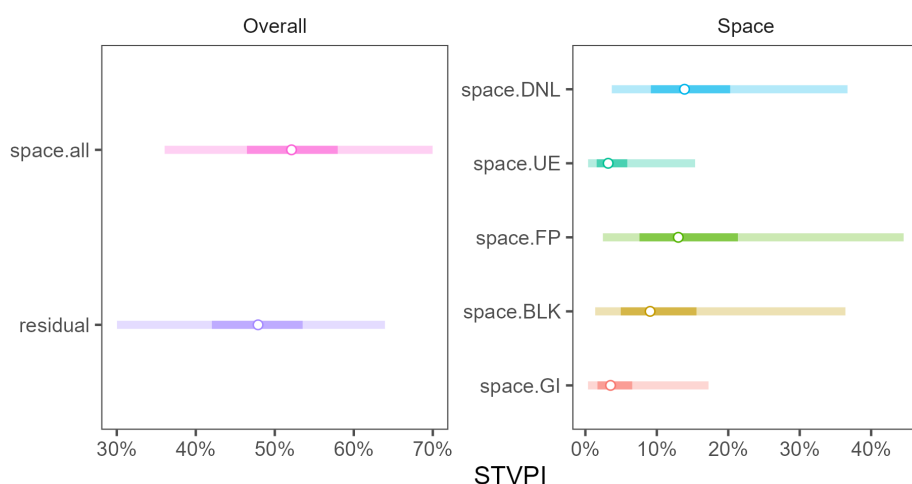
# 提取制图数据
## 此例为连续型结果的制图，若为计数型或二值型，需删除 "space.time.all","residual" 字段
random.effects.BSVC <- c(
  "space.all", "residual", # 模型可解释度与残差
  "space.DNL", "space.UE", "space.FP", "space.BLK", "space.GI"
) # 每个 X 的空间贡献百分比
# 筛选数据并对数据进行分组，便于后续制图
plot.data.BSVC <- as.data.frame(STVPI.data.BSVC[STVPI.data.BSVC$`random effects` %in%
  random.effects.BSVC, ]) %>%

  mutate(Group = case_when(
    `random effects` %in% random.effects.BSVC[1:2] ~ "Overall",
    `random effects` %in% random.effects.BSVC[3:7] ~ "Space"
  )) %>%
  filter(Group != "NA")

# 逆转因子的展示顺序
plot.data.BSVC$`random effects` <- factor(plot.data.BSVC$`random effects`,
  levels = rev(random.effects.BSVC))

# 绘制 STVPI 图
(p9 <- ggplot(plot.data.BSVC, aes(x = STVPI_mean, y = `random effects`, group = Group)) +
  geom_errorbarh(aes(xmin = `STVPI_2.5%`, xmax = `STVPI_97.5%`,
    colour = `random effects`),
    height = 0, linewidth = 1.8, alpha = 0.3, linetype = "solid") +
  geom_errorbarh(aes(xmin = `STVPI_25%`, xmax = `STVPI_75%`,
    colour = `random effects`),
    height = 0, linewidth = 1.8, alpha = 0.6, linetype = "solid") +
  geom_point(aes(colour = `random effects`), shape=21, fill="white", size =1.8,alpha=1) +
  labs(title = "", x = "STVPI", y = "") +
  facet_wrap(~Group, scale = "free", ncol = 3) +
  theme_few() +
  scale_x_continuous(labels = scales::percent_format()) +
  theme(legend.position = "none"))
```





【注意】泊松-BSVC（计数型）和 logistic-BSVC（二值型）模型目前也未包括残差项，因此同样需删除 `random.effects.BSVC` 变量中的“space.time.all”, “residual” 字段，以及 `plot.data.BSVC` 变量中的“Overall” 字段。

#### 4.5.4 空间回归系数（空间非平稳）

**4.5.4.1 绘制空间回归系数地图** 提取出模型输出中的空间回归系数数据框后，建议在 ArcGIS 或 ArcGIS Pro 等专业地图制图软件中进行绘制。此处基于 R 语言，简单绘制出基础的空间回归系数地图展示结果，仅供样本制图参考。

```
# 提取模型输出中的空间回归系数
Space.Coeff.BSVC <- model_type1$space.coefficients

# 组合地图中的几何属性与每个地理单元的空间回归系数
Space.Coeff.BSVC.merge <- merge(Florida_Map, Space.Coeff.BSVC, by = "FIPS", all.x = T)

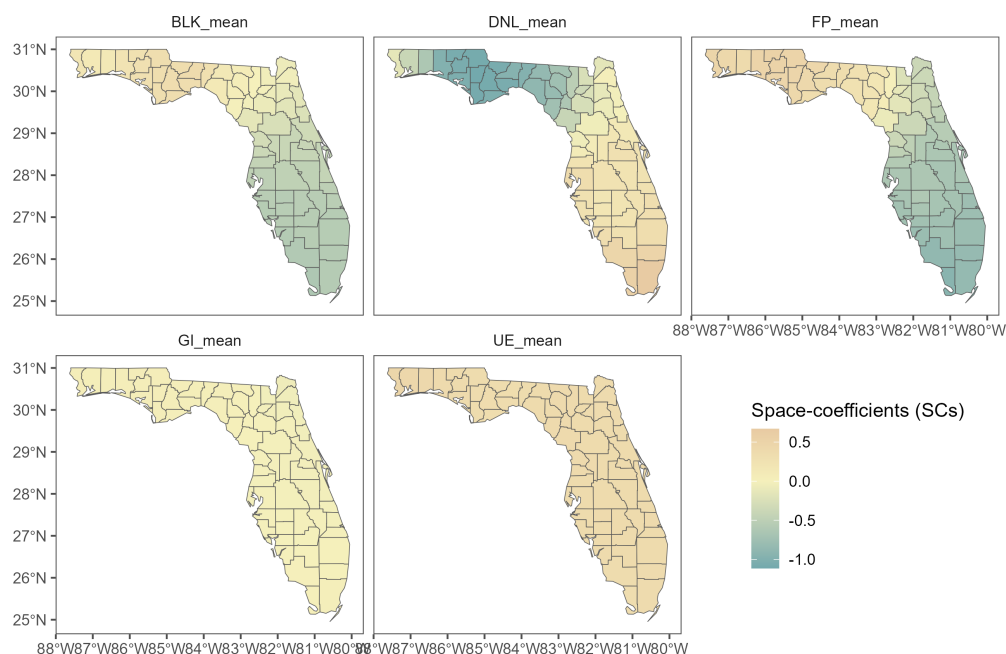
# 将截面数据转换为面板数据格式
Space.coef.BSVC.panel <- Space.Coeff.BSVC.merge %>%
  pivot_longer(
    cols = paste0(c("DNL", "UE", "FP", "BLK", "GI"), "_mean"),
    names_to = "variable",
    values_to = "SCs"
  ) %>%
  select(c(1:9, 45:47))

# 绘制地图
(p10 <- ggplot() +
  geom_sf(data = Space.coef.BSVC.panel, aes(group = variable, fill = SCs)) +
  facet_wrap(~variable, shrink = F, drop = F) +
```

```

scale_fill_gradient2(low = "#73a9ad", mid = "#f5f0bb", high = "#dfb293",
                     name = "Space-coefficients (SCs)") +
theme_few() +
theme(legend.position = c(.85, .25))
)

```



**4.5.4.2 单个空间单元的因子影响效应制图** 仿照 SHAP 制图范式，本图展示了单个空间单元的因子影响效应图。该图为两类图形的组合：

- 横向柱形图：每个横向柱形代表了某个空间单元的不同解释变量  $X$  的空间回归系数；
- 线段图：叠加在柱形图上的线段表示了这些回归系数的宽（95%）和窄（50%）贝叶斯可信区间。

这种可视化方法有助直观地理解不同空间单元对目标变量影响的空间异质性，以及估计系数的不确定性范围。以下是绘制单个空间单元的空间回归系数的 R 代码，批量绘制多个空间单元的因子影响效应图的 R 代码可参考此小节的后部分。

```

# 提取 X 名字
name.X.BSVC <- gsub("_mean", "", names(Space.Coeff.BSVC[, c(2, 10, 18, 26, 34)]))

# 手动处理数据 # 转换为制图数据（时空面板数据格式）
SC.BSVC <- list()
for (name in c("mean", "0.025quant", "0.25quant", "0.75quant", "0.975quant")) {
  SC.BSVC[[name]] <- Space.Coeff.BSVC %>%
    pivot_longer(cols = paste0(c("DNL", "UE", "FP", "BLK", "GI"), "_", name),
                 names_to = "feature", values_to = name)
}

```

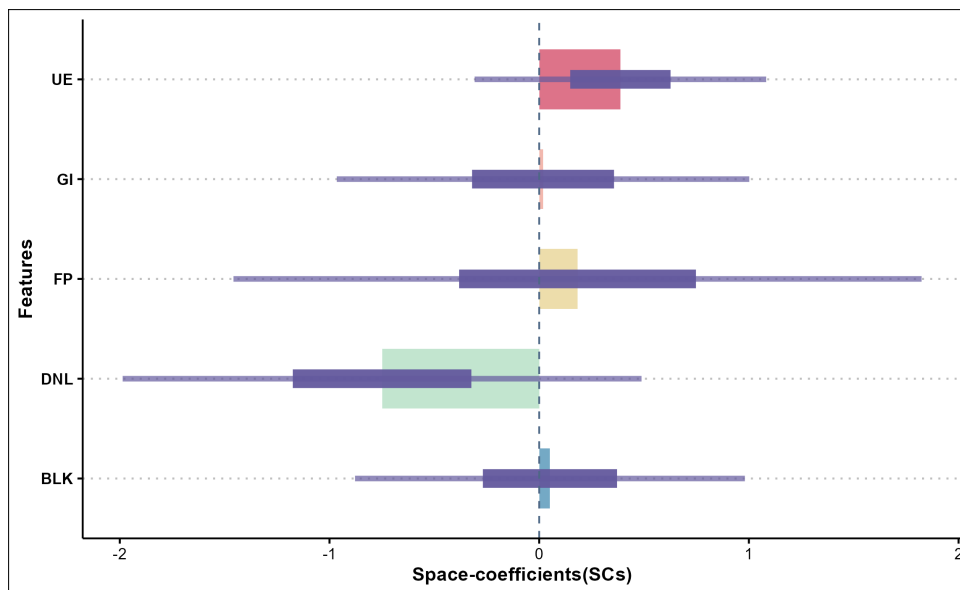
```

}

# 组合制图数据
impact_data_BSVC <- cbind.data.frame(
  feature = rep(name.X, times = nrow(Space.Coeff.BSVC)), # 变量
  SC.BSVC$mean$FIPS,
  SC.BSVC$mean$mean, SC.BSVC$`0.025quant`$`0.025quant`, SC.BSVC$`0.25quant`$`0.25quant`,
  SC.BSVC$`0.75quant`$`0.75quant`, SC.BSVC$`0.975quant`$`0.975quant`
) %>%
  `colnames<-`(c("feature", "FIPS", "mean", "0.025quant", "0.25quant",
    "0.75quant", "0.975quant"))

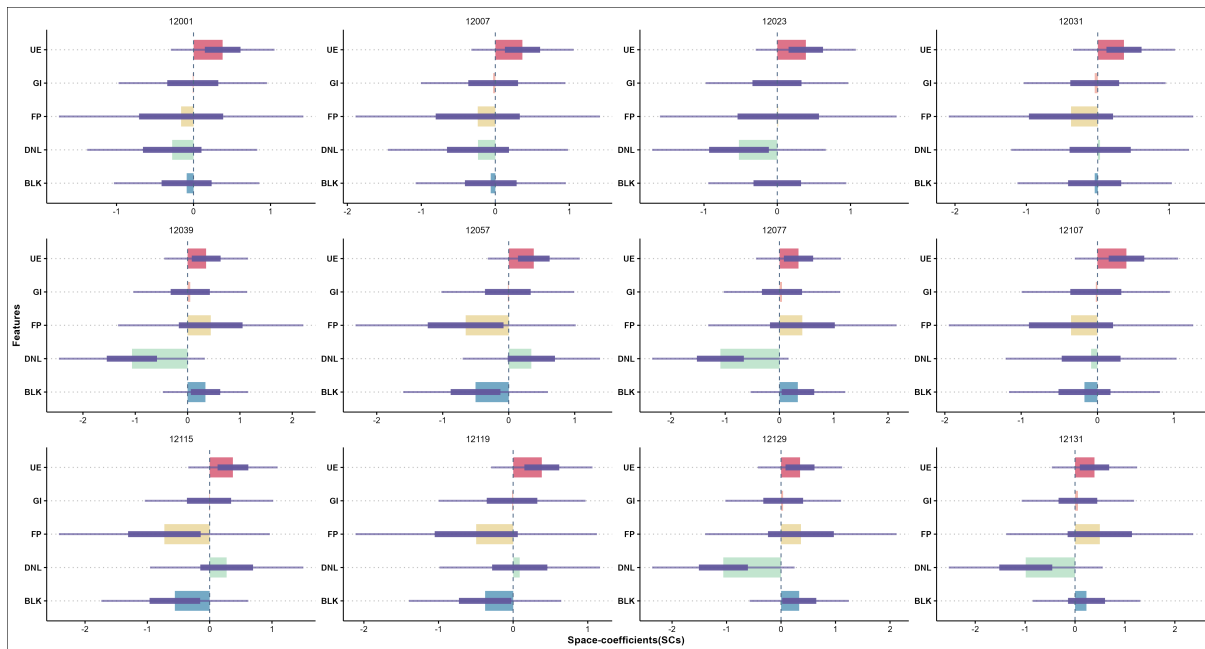
## 绘制某个空间单元的因子影响效应
# 提取唯一值字段为 "12121" 的空间回归系数
single_data_BSVC <- impact_data_BSVC %>% filter(FIPS == 12121)
# 绘图
(p11 <- ggplot(data = single_data_BSVC)+
  geom_bar(
    aes(x = mean, y = feature, fill = feature),
    stat = "identity", width = 0.6)+
  geom_segment(aes(x = `0.025quant`, y = feature,
    xend = `0.975quant`, yend = feature),
    color="#63599d",size = 1.5,alpha=.7)+
  geom_segment(aes(x = `0.25quant`, y = feature,
    xend = `0.75quant`, yend = feature),
    color="#63599d",size = 5,alpha=.95)+
  # 添加水平参考线
  geom_vline(xintercept = 0, linetype = "dashed", size = 0.5, color = "#4e6886") +
  scale_fill_manual(values = c("#74a9c5", "#c2e5cf", "#edddab", "#f2b8ae", "#dd7389")) +
  scale_color_manual(values = c("#74a9c5", "#c2e5cf", "#edddab", "#f2b8ae", "#dd7389")) +
  theme_clean()+
  labs(x="Space-coefficients(SCs)",y="Features")+
  theme(
    axis.text.y = element_text(size = 9, face = "bold"),
    axis.title.x = element_text(size = 11, face = "bold"),
    axis.title.y = element_text(size = 11, face = "bold"),
    legend.position = "none")
)

```



```
## 绘制多个空间单元的因子影响效应
# 随机提取 12 个空间单元的空间回归系数
multi_data_BSVC <- impact_data_BSVC[impact_data_BSVC$FIPS %in%
                                     sample(unique(impact_data_BSVC$FIPS),12),]

# 绘图
(p12 <- ggplot(data = multi_data_BSVC,
               aes(x = mean, y = feature, fill = feature))+
  geom_bar(stat = "identity", width = 0.6)+
  geom_segment(aes(x = `0.025quant`, y = feature,
                  xend = `0.975quant`, yend = feature),
              color="#63599d",size = 1,alpha=.7)+
  geom_segment(aes(x = `0.25quant`, y = feature,
                  xend = `0.75quant`, yend = feature),
              color="#63599d",size = 3,alpha=.95)+
  # 添加水平参考线
  geom_vline(xintercept = 0, linetype = "dashed", size = 0.5, color = "#4e6886") +
  scale_fill_manual(values = c("#74a9c5", "#c2e5cf", "#edddab", "#f2b8ae", "#dd7389")) +
  scale_color_manual(values = c("#74a9c5", "#c2e5cf", "#edddab", "#f2b8ae", "#dd7389")) +
  facet_wrap(~FIPS, scale="free",ncol=4)+
  theme_clean()+
  labs(x="Space-coefficients(SCs)",y="Features")+
  theme(
    axis.text.y = element_text(size = 9, face = "bold"),
    axis.title.x = element_text(size = 11, face = "bold"),
    axis.title.y = element_text(size = 11, face = "bold"),
    legend.position = "none")
)
```



**4.5.4.3 空间回归系数的蜂巢-条状组合图** 仿照 SHAP 制图范式，本图展示了模型中各个变量的空间回归系数分布情况。每一行代表一个不同的解释变量，而每个点则对应于该变量在每个空间单元上的空间回归系数：

- 点的分布：图中的散点表示各个空间单元的空间回归系数，这些点的分布揭示了变量在空间上的影响程度和方向的变异性。
- 平均值柱形图：每个变量下方的柱形图显示了该变量所有空间回归系数的平均值（即代表全局空间回归系数），柱形图的颜色深浅表示平均系数的影响方向，从而提供了对变量整体影响的直观理解。

# 手动创建直方图数据

```
bar_data_BSVC <- data.frame(
  feature = rep(name.X.BSVC, each = nrow(Space.Coeff.BSVC)), # 变量
  coef_value = c(Space.Coeff.BSVC$DNL_mean, Space.Coeff.BSVC$UE_mean, Space.Coeff.BSVC$FP_mean,
    Space.Coeff.BSVC$BLK_mean, Space.Coeff.BSVC$GI_mean) # 局部空间回归系数
) %>%
group_by(feature) %>%
summarise(
  mean = mean(coef_value),
  abs_mean = mean(abs(coef_value))
)
```

# 手动创建蜂窝图数据

```
beeswarm_data_BSVC <- data.frame(
  feature = rep(name.X.BSVC, each = nrow(Space.Coeff.BSVC)), # 变量
  coef_value = c(Space.Coeff.BSVC$DNL_mean, Space.Coeff.BSVC$UE_mean, Space.Coeff.BSVC$FP_mean,
```

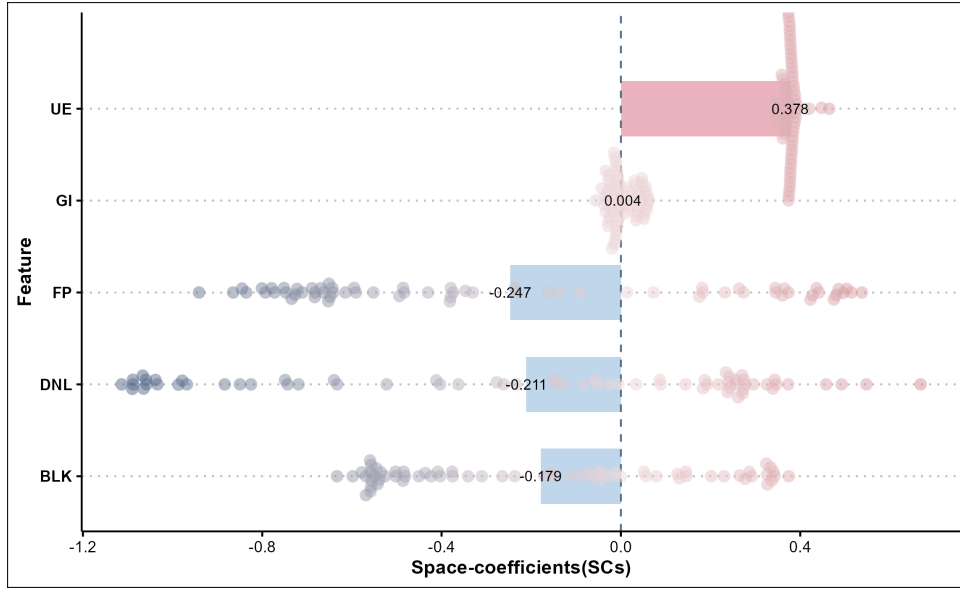
```

        Space.Coeff.BSVC$BLK_mean, Space.Coeff.BSVC$GI_mean) # 局部空间回归系数
    )

# 绘制组合图
(p13 <- ggplot() +
  ## 绘制平均的空间回归系数（全局空间回归系数）
  geom_bar(
    stat = "identity", data = bar_data_BSVC,
    aes(x = mean, y = feature, fill = mean < 0), width = 0.6
  ) +
  scale_fill_manual(values = c("#EBB3BE", "#C0D6EA")) +
  # 添加水平参考线
  geom_vline(xintercept = 0, linetype = "dashed", size = 0.5, color = "#4e6886") +

  ## 绘制局部空间回归系数分布图
  geom_beeswarm(
    data = beeswarm_data_BSVC,
    aes(x = coef_value, y = feature, colour = coef_value),
    alpha = 0.5, cex = 1, shape = 16, size = 3, priority = "descending"
  ) +
  scale_colour_gradient2(low = "#4e6886", mid = "#eed8db", high = "#bb616d",
    midpoint = 0, name = NULL) + # 渐变颜色
  geom_text(
    data = bar_data_BSVC, aes(x = mean, y = feature, label = round(mean, 3)),
    # position = position_stack(vjust = 30, reverse = T),
    size = 3,
    color = "black"
  ) + # 在柱形上添加数值
  theme_clean() +
  theme(
    plot.title = element_text(hjust = 0.5, color = "purple"),
    axis.text.y = element_text(size = 9, face = "bold"),
    axis.title.x = element_text(size = 11, face = "bold"),
    axis.title.y = element_text(size = 11, face = "bold"),
    legend.position = "none"
  ) +
  labs(
    title = NULL,
    x = "Space-coefficients(SCs)",
    y = "Feature"
  ))

```



## 5 BSTVC 的引用文献

- [Bayesian STVC series models] Song, Chao, Yin, Hao, Shi, Xun, Xie, Mingyu, Yang, Shujuan, Zhou, Junmin, Wang, Xiuli, Tang, Zhangying, Yang, Yili, & Pan, Jay. (2022). Spatiotemporal disparities in regional public risk perception of COVID-19 using Bayesian Spatiotemporally Varying Coefficients (STVC) series models across Chinese cities. *International Journal of Disaster Risk Reduction*, 77, 103078.
- [STVPI] Wan, Qin, Tang, Zhangying, Pan, Jay, Xie, Mingyu, Wang, Shaobin, Yin, Hao, Li, Junmin, Liu, Xin, Yang, Yang, & Song, Chao. (2022). Spatiotemporal heterogeneity in associations of national population ageing with socioeconomic and environmental factors at the global scale. *Journal of Cleaner Production*, 373, 133781.
- Song, Chao, Shi, Xun, & Wang, Jinfeng. (2020). Spatiotemporally Varying Coefficients (STVC) model: a Bayesian local regression to detect spatial and temporal nonstationarity in variables relationships. *Annals of GIS*, 26(3), 277-291.
- Song, Chao, Shi, Xun, Bo, Yanchen, Wang, Jinfeng, Wang, Yong, & Huang, Dacang. (2019). Exploring Spatiotemporal Nonstationary Effects of Climate Factors on Hand, Foot, and Mouth Disease Using Bayesian Spatiotemporally Varying Coefficients (STVC) Model in Sichuan, China. *Science of The Total Environment*, 648, 550-560.

## 6 应用案例

- Song, Chao, Fang, Lina, Xie, Mingyu, Tang, Zhangying, Zhang, Yumeng, Tian, Fan, Wang, Xiuli, Lin, Xiaojun, Liu, Qiaolan, Xu, Shixi, & Pan, Jay. (2024). Revealing spatiotemporal inequalities, hotspots, and determinants in healthcare resource distribution: insights from hospital beds panel data in 2308 Chinese counties. *BMC Public Health*, 24, 423.



- Song, Chao, Wang, Xiuli, Ge, Erjia, Shi, Xun, & Pan, Jay. (2024). Applications of Geospatial Information Technologies and Spatial Statistics in Health Services Research. *Frontiers in Public Health*, 11, 1349985.
- Zhang, Xu, Song, Chao, Wang, Chengwu, Yang, Yili, Ren, Zhoupeng, Xie, Mingyu, Tang, Zhangying, & Tang, Honghu. (2021). Socioeconomic and Environmental Impacts on Regional Tourism across Chinese Cities: A Spatiotemporal Heterogeneous Perspective. *ISPRS International Journal of Geo-Information*, 10(6), 410.
- Song, Chao, Wang, Yaode, Yang, Xiu, Yang, Yili, Tang, Zhangying, Wang, Xiuli, & Pan, Jay. (2020). Spatial and Temporal Impacts of Socioeconomic and Environmental Factors on Healthcare Resources: A County-Level Bayesian Local Spatiotemporal Regression Modeling Study of Hospital Beds in Southwest China. *International Journal of Environmental Research and Public Health*, 17, 5890.
- Yang, Yang, Yang, Jintao, Xu, Chengdong, Xu, Chong, & Song, Chao. (2019). Local-scale landslide susceptibility mapping using the B-GeoSVC model. *Landslides*, 16(7), 1301-1312.
- 罗雅玲, 唐欣怡, 解铭宇, 严国强, 唐先腾, 唐章英, 王秀丽, 潘杰, & 宋超. (2025). 医疗资源配置效率的小区域时空评价: 基于四川省县域面板数据的实证研究. *中国卫生事业管理*.

## 7 Q & A

1. **关于计算复杂度:** 由于 BSTVC 模型采用全地图单独建模框架, 计算强度较大, 因此需要较高内存配置的计算机平台, 同时运算时间较长。
2. **BSTVC 与频率统计下 GTWR 的区别:** 参照“BSTVC 模型的优势”, 值得一提的是, BSTVC 采用了时空分离的非平稳假设, 而 GTWR 则采用时空交互的非平稳假设, 具体区别可参见 [Song et al. 2020](#) 中的相关论述。此外, 基于时空交互假设的模型是 BSTIVC ([Song et al., 2022](#)), 该模型目前尚未包括在本 R 包中。
3. **BSTVC 与 SHAP 的区别:** SHAP (SHapley Additive exPlanations) 是一个用于揭示特征对模型输出贡献的工具, 主要解释局部回归系数。然而, 与 BSTVC 模型相比, SHAP 在某些方面存在局限性: (1) **不确定性量化:** BSTVC 提供参数估计的不确定性量化, 包含 95% 和 50% 的贝叶斯可信区间, 能够为预测和推断提供合理的统计范围, 而 SHAP 无法实现这一点; (2) **时空自相关:** BSTVC 考虑时空数据中的自相关性, 并在“全地图”框架下统一拟合局部回归系数, 确保参数估算的准确性和可靠性, 而 SHAP 无法处理时空自相关, 并采用“分离单一”建模方式, 缺乏全局一致性。
4. **贝叶斯 STVC 系列模型:** 目前该系列包括四种子模型 (STVI、STIVI、STVC、STIVC), 旨在解决时空交互效应和时空描述型分析等更复杂场景的需求。后续版本将发布更多功能。
5. **关于空间权重矩阵:** k 临近方法可能会报错或无法输出结果, 这可能与原始贝叶斯推断算法的实现有关。
6. **关于残差项:** 泊松-BSTVC 和 logistic-BSTVC 模型目前未包括残差项, 因此 STVPI 无法计算整体模型的可解释百分比。后续版本将推出此功能。

7. 关于时空非平稳：当前版本的 BSTVC 函数假设所有解释变量均存在时空非平稳性，后续版本将推出支持单个解释变量无时间非平稳性假设的更灵活模型。

## 8 版本更新与技术支持

BSTVC 包将会不定期更新以引入新功能和改进。我们欢迎并鼓励用户贡献，包括报告问题、提出功能请求或提交代码更改。如果您在使用 BSTVC 包时遇到任何问题，或需要进一步的帮助，可以通过以下方式获取支持：

- 用户手册：提供详细的模型描述、安装指南和示例，中文请访问[GetStart-Chinese](#)进行下载，英文请访问[GetStart-English](#)进行下载。
- GitHub Issues：在 GitHub 仓库中报告问题或请求新功能，请访问[Issues](#)。
- 邮件联系：tangxxxxt@163.com (唐先腾，R 包使用相关); chaosong.gis@gmail.com (宋超，统计理论相关)
- Bayesian STVC model: <https://chaosong.blog/bayesian-stvc/>].

版权 © HEOA-华西健康医学地理课题组

微信公众号：医学地理信息与空间卫生统计/HealthGeography

