## **Tidyverse**

是一组处理与可视化R包的集合，其中ggplot2与dplyr最广为人知。<https://www.jianshu.com/p/f3c21a5ad10a>

核心包有以下一些：

ggplot2 - 可视化数据

dplyr - 数据操作语法，可以用它解决大部分数据处理问题

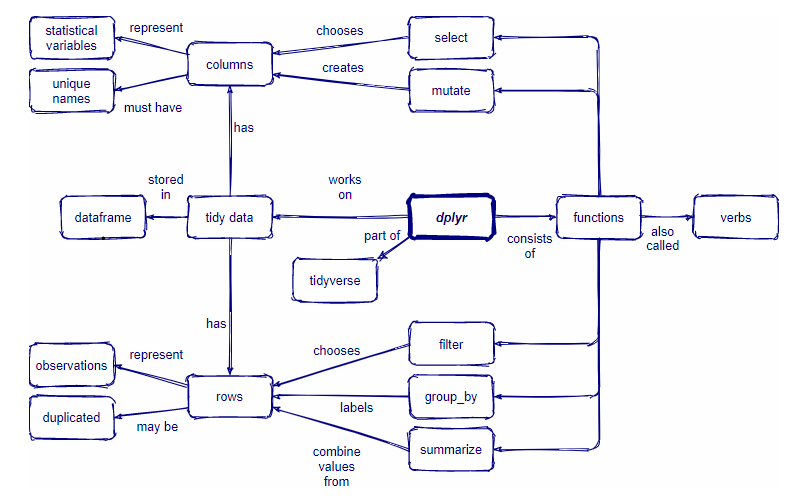
tidyr - 清理数据

readr - 读入表格数据

tibble - 新一代数据框

stringr - 提供函数集用来处理字符数据

forcats - 提供有用工具用来处理因子Dplyr



#### filter(.data, ..., .preserve = FALSE)

对数据帧进行子集化

BoroughMap <- LondonBoroughs %>%

dplyr::filter(str\_detect(GSS\_CODE, "^E09"))

#### dplyr::count()

按组计数观察结果

Qcount <- BluePlaquesSub.ppp %>%

quadratcount(.,nx = 6, ny = 6) %>%

as.data.frame() %>%

dplyr::count(Var1=Freq)

#### mutate()

创建、修改和删除列

sums <- Qcount %>%

mutate(total = col\_a + col\_b)

#### rename(.data, ...)

对列重命名

Qcount <- dplyr::rename(new\_colname=old\_colname)

#### distinct(.data, ..., .keep\_all = FALSE)

对data frame删除重复行。如果不在…中指明保留哪些列，默认.keep all=FALSE，不保留所有列，.keep all=TRUE保留所有列。

LondonWardsMerged <-distinct(LondonWardsMerged,GSS\_CODE, ward\_name)

#### pull(.data, var = -1, name = NULL, ...)

提取单个列

I\_LWard\_Global\_Density <- points\_sf\_joined %>%

pull(density)

## sf

提供简单访问功能的包，simple feature表示为带有几何列表列的data.frame或tibble中的记录

<https://github.com/r-spatial/sf>

#### st\_read(dsn, layer, ...)

从文件或数据库读取simple feature

LondonBoroughs <- st\_read(here::here("ESRI", "London\_Borough\_Excluding\_MHW.shp"))

#### st\_transform(x, crs, ...)

对数据帧进行子集化，保留满足条件的所有行

BoroughMap <- LondonBoroughs %>%

dplyr::filter(str\_detect(GSS\_CODE, "^E09"))%>%

st\_transform(., 27700) #选取GSS\_CODE列带有E09字符的行，并转换坐标系为27700

#### st\_join(x, y, join, ...)

空间连接，空间过滤

points\_sf\_joined <- st\_join(LondonWardsMerged,BluePlaquesSub)

#### str\_detect(string, pattern, negate = FALSE)

检测字符串中是否存在要求的string。

str\_detect(GSS\_CODE, "^E09")

#### st\_bbox()

返回一个边界的simple feature

HarrowWGSbb <- Harrow %>%

st\_bbox()

## tmap

<http://wap.sciencenet.cn/blog-255662-1144743.html>

#### qtm()

快速专题地图图

qtm(

shp, fill = NA,

symbols.size = NULL, symbols.col = NULL, symbols.shape = NULL,

dots.col = NULL,

text = NULL, text.size = 1, text.col = NA,

lines.lwd = NULL, lines.col = NULL,

raster = NA, borders = NA,

scale = NA, title = NA,

projection = NULL,

bbox = NULL, basemaps = NA,

overlays = NA, style = NULL, format = NULL)

#### tmap\_mode()

tmap\_mode(mode = c("plot", "view"))

将tmap模式设置为静态绘图或交互式查看，之后绘制地图

tmap\_mode("view")

tm\_shape(BoroughMap) +

tm\_polygons(col = NA, alpha = 0.5) +

tm\_shape(BluePlaques) +

tm\_dots(col = "blue")

## spatstat

为空间点模式做统计分析的R包

#### density(x, sigma, ..., edge=TRUE,

**method=c("FFT", "C", "interpreted"),**

**at=NULL)**

点核密度平滑，用于Kernel Density Estimation，sigma设置kernel半径

BluePlaquesSub.ppp %>%

density(., sigma=500)

#### quadratcount(X, nx=5, ny=nx, ...,

**xbreaks=NULL, ybreaks=NULL, tess=NULL)**

点模式的样方计数

BluePlaquesSub.ppp %>%

quadratcount(.,nx = 6, ny = 6)

#### as.data.frame(x, row.names = NULL, ...)

强制让ppp class的点pattern转换到到data frame

Qcount <- BluePlaquesSub.ppp %>%

quadratcount(.,nx = 6, ny = 6) %>%

as.data.frame()

#### quadrat.test()

基于样方计数的空间点模式的离散检验

teststats <- quadrat.test(BluePlaquesSub.ppp, nx = 6, ny = 6)

#### kest()

对数据进行Ripley’s K test

K <- BluePlaquesSub.ppp %>%

Kest(., correction="border")

## DBSCAN

#### kNNdistplot(x, k = 4, ...)

计算并绘制k近邻距离

BluePlaquesSubPoints%>%

dbscan::kNNdistplot(.,k=4)

## fpc

#### fpc:: dbscan(x, eps, minPts = 5, weights = NULL, borderPoints = TRUE, ...)

使用kd-tree的DBSCAN(Density-based spatial clustering of applications with noise)聚类算法

db <- BluePlaquesSubPoints %>%

fpc::dbscan(.,eps = 700, MinPts = 4)

## ggplot2

#### aes(x, y, ...)

设置地图展示效果

dbplot <- dbplot + geom\_polygon(data = chulls,

aes(coords.x1,coords.x2, group=dbcluster),

alpha = 0.5)

#### ggplot(data = NULL, mapping = aes(), ..., environment = parent.frame())

建立一个新的ggplot

dbplot <- dbplot + geom\_polygon(data = chulls,

mapping=aes(coords.x1,coords.x2, group=dbcluster),

alpha = 0.5)