

R 地图可视化

Code ▼

北京各区二手房价格及电梯配置分析

Hide

```
library(sf)
```

```
Linking to GEOS 3.13.1, GDAL 3.10.2, PROJ 9.5.1; sf_use_s2() is TRUE
```

Hide

```
library(ggplot2)
library(dplyr)
```

```
载入程序包： ‘dplyr’

The following objects are masked from ‘package:stats’ :

  filter, lag

The following objects are masked from ‘package:base’ :

  intersect, setdiff, setequal, union
```

Hide

```
library(ggforce)
library(tidyr)
library(ggspatial)
library(ggnewscale)

# 1. 检查地理数据字段
bj_shp <- st_read("北京市.shp") %>%
  st_make_valid() %>%
  st_simplify(dTolerance = 0.01)
```

```
Reading layer `北京市` from data source `D:\data_analysis\R语言可视化\北京市.shp` using driver
`ESRI Shapefile`
Simple feature collection with 16 features and 9 fields
Geometry type: MULTIPOLYGON
Dimension:      XY
Bounding box:   xmin: 115.4234 ymin: 39.44276 xmax: 117.5146 ymax: 41.0608
Geodetic CRS:   WGS 84
```

Hide

```
# 查看地理数据的字段名称和示例值
cat("地理数据字段名称:\n")
```

地理数据字段名称：

Hide

```
names(bj_shp)
```

```
[1] "adcode"      "name"        "center"      "centroid"    "childrenNu"  "level"      "parent"
"subFeature"  "acroutes"    "geometry"
```

Hide

```
cat("\n行政区划名称示例:\n")
```

行政区划名称示例：

Hide

```
head(bj_shp$NAME)
```

```
NULL
```

Hide

```
# 2. 检查房屋数据字段
house_data <- read.csv("北京_geo.csv", fileEncoding = "GBK") %>%
  filter(总价.万元. > 0) %>%
  mutate(配备电梯 = ifelse(配备电梯 == "有", "有电梯", "无电梯"))

cat("\n房屋数据区字段示例:\n")
```

房屋数据区字段示例：

Hide

```
head(unique(house_data$区))
```

```
[1] "顺义" "怀柔" "昌平" "房山" "大兴" "密云"
```

Hide

```
# 3. 统一行政区划名称（需要根据实际数据调整）
# 常见需要处理的名称差异示例：
# 地理数据：“东城区” 房屋数据：“东城”
# 解决方法：
bj_shp <- bj_shp %>%
  mutate(district = gsub("区|市辖区", "", name))

house_data <- house_data %>%
  mutate(区 = gsub("区", "", 区))

# 4. 重新执行数据合并
district_avg <- house_data %>%
  group_by(区) %>%
  summarise(平均总价 = mean(总价.万元., na.rm = TRUE))

elevator_ratio <- house_data %>%
  count(区, 配备电梯) %>%
  group_by(区) %>%
  mutate(比例 = n / sum(n)) %>%
  ungroup()

# 使用统一后的字段名合并
bj_shp <- bj_shp %>%
  left_join(district_avg, by = c("district" = "区")) %>%
  left_join(elevator_ratio, by = c("district" = "区"))

# 5. 最终可视化代码
centers <- bj_shp %>%
  st_centroid() %>%
  st_coordinates() %>%
  as.data.frame() %>%
  rename(center_x = X, center_y = Y)
```

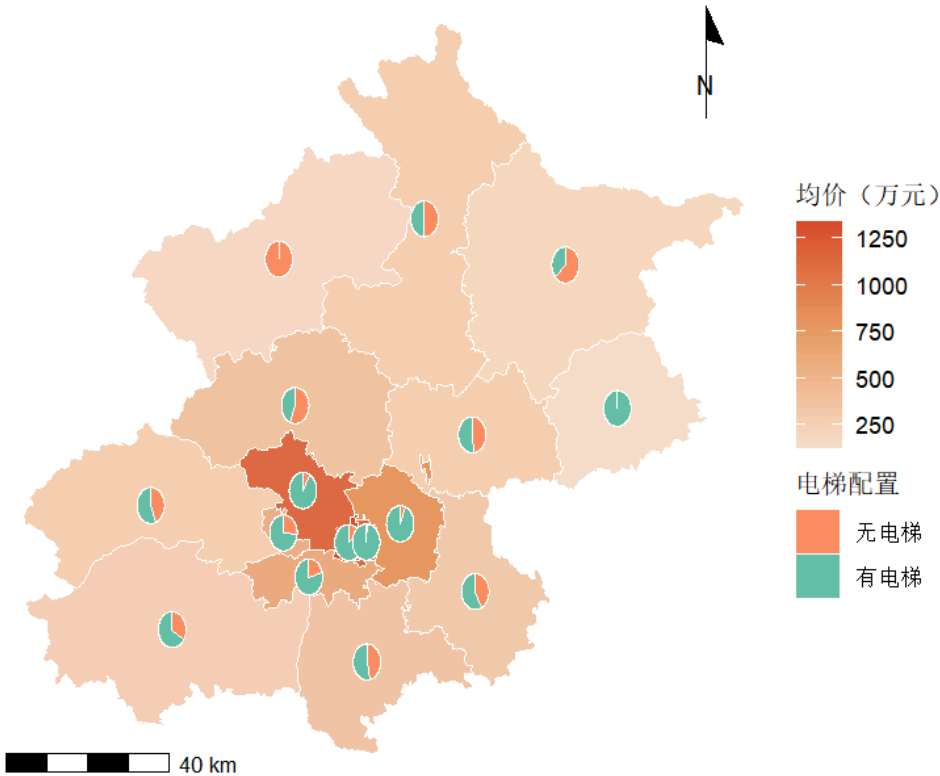
警告：st_centroid assumes attributes are constant over geometries

[Hide](#)

```
ggplot() +  
  geom_sf(data = bj_shp, aes(fill = 平均总价), color = "white", size = 0.2) +  
  scale_fill_gradientn(  
    name = "均价（万元）",  
    colors = c("#F7DFCC", "#E89F67", "#D94B2B"),  
    na.value = "grey90"  
  ) +  
  ggnewscale::new_scale_fill() +  
  geom_arc_bar(  
    data = bj_shp %>% st_drop_geometry() %>% drop_na(配备电梯),  
    aes(x0 = centers$center_x,  
        y0 = centers$center_y,  
        r0 = 0,  
        r = 0.04,  
        amount = 比例,  
        fill = 配备电梯),  
    stat = "pie",  
    color = "white",  
    size = 0.06  
  ) +  
  scale_fill_manual(  
    name = "电梯配置",  
    values = c("有电梯" = "#66C2A5", "无电梯" = "#FC8D62")  
  ) +  
  annotation_scale() +  
  annotation_north_arrow(  
    location = "tr",  
    style = north_arrow_minimal()  
  ) +  
  labs(title = "北京市各区二手房价格及电梯配置分析") +  
  theme_void() +  
  theme(legend.position = "right",  
        plot.title = element_text(hjust = 0.5, face = "bold"))
```

警告: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
Please use `linewidth` instead.

北京市各区二手房价格及电梯配置分析



北京各区二手房单价及装修情况分析

Hide

```
library(sf)
library(ggplot2)
library(dplyr)
library(ggforce)
library(tidyr)
library(ggspatial)
library(ggnewscale)

# 1. 读取地图和房屋数据
bj_shp <- st_read("北京市.shp") %>%
  st_make_valid() %>%
  st_simplify(dTolerance = 0.01)
```

Reading layer `北京市` from data source `D:\data_analysis\R语言可视化\北京市.shp` using driver `ESRI Shapefile`
Simple feature collection with 16 features and 9 fields
Geometry type: MULTIPOLYGON
Dimension: XY
Bounding box: xmin: 115.4234 ymin: 39.44276 xmax: 117.5146 ymax: 41.0608
Geodetic CRS: WGS 84

Hide

```
house_data <- read.csv("北京_geo.csv", fileEncoding = "GBK") %>%
  filter(总价.万元. > 0) %>%
  mutate(
    配备电梯 = ifelse(配备电梯 == "有", "有电梯", "无电梯"),
    区 = gsub("区", "", 区)
  )

# 2. 标准化地理数据区字段
bj_shp <- bj_shp %>%
  mutate(district = gsub("区|市辖区", "", name))

# 3. 计算每区的平均单价和装修比例
district_avg <- house_data %>%
  group_by(区) %>%
  summarise(平均单价 = mean(单价.元.平., na.rm = TRUE))

decoration_ratio <- house_data %>%
  count(区, 装修情况) %>%
  group_by(区) %>%
  mutate(比例 = n / sum(n)) %>%
  ungroup()

# 4. 合并地理与房屋数据
bj_shp <- bj_shp %>%
  left_join(district_avg, by = c("district" = "区"))

# 获取中心点坐标
centers <- bj_shp %>%
  st_centroid() %>%
  st_coordinates() %>%
  as.data.frame() %>%
  rename(center_x = X, center_y = Y)
```

警告: st_centroid assumes attributes are constant over geometries

[Hide](#)

```

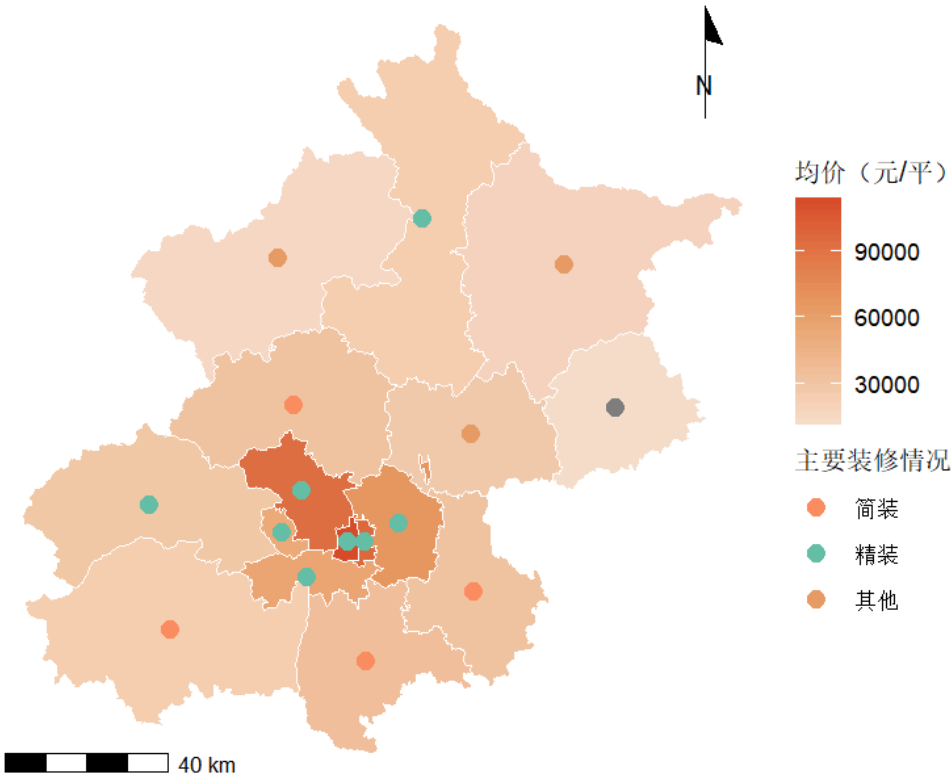
# 装修情况点图数据（按装修最多的类型代表该区）
top_decoration <- house_data %>%
  group_by(区, 装修情况) %>%
  summarise(n = n(), .groups = "drop") %>%
  group_by(区) %>%
  slice_max(n, n = 1) %>%
  rename(district = 区)

# 与中心点坐标合并
decoration_points <- bj_shp %>%
  st_drop_geometry() %>%
  select(district) %>%
  bind_cols(centers) %>%
  left_join(top_decoration, by = "district") %>%
  drop_na(装修情况)

# 5. 可视化
ggplot() +
  geom_sf(data = bj_shp, aes(fill = 平均单价), color = "white", size = 0.2) +
  scale_fill_gradientn(
    name = "均价（元/平）",
    colors = c("#F7DFCC", "#E89F67", "#D94B2B"),
    na.value = "grey90"
  ) +
  ggnewscale::new_scale_color() +
  geom_point(
    data = decoration_points,
    aes(x = center_x, y = center_y, color = 装修情况),
    size = 3
  ) +
  scale_color_manual(
    name = "主要装修情况",
    values = c("精装" = "#66C2A5", "简装" = "#FC8D62", "其他" = "#E89F67")
  ) +
  annotation_scale() +
  annotation_north_arrow(
    location = "tr",
    style = north_arrow_minimal()
  ) +
  labs(title = "北京市各区二手房单价与主要装修情况分布图") +
  theme_void() +
  theme(
    legend.position = "right",
    plot.title = element_text(hjust = 0.5, face = "bold")
  )

```

北京市各区二手房单价与主要装修情况分布图



Hide

NA
NA

上海各区二手房价格及电梯配置分析

Hide

```
library(sf)
library(ggplot2)
library(dplyr)
library(ggforce)
library(tidyr)
library(ggspatial)
library(ggnewscale)

# 1. 数据准备 -----
# 读取上海市的行政区边界数据并进行预处理
sh_shp <- st_read("上海市.shp") %>%
  st_make_valid() %>% # 确保几何数据有效
  st_simplify(dTolerance = 0.01) %>% # 简化几何形状，提高渲染效率
  mutate(district = gsub("区|市辖区", "", name)) # 去除区和市辖区的名称
```

Reading layer `上海市` from data source `D:\data_analysis\R语言可视化\上海市.shp` using driver `ESRI Shapefile`
Simple feature collection with 16 features and 9 fields
Geometry type: MULTIPOLYGON
Dimension: XY
Bounding box: xmin: 120.8568 ymin: 30.67559 xmax: 122.2471 ymax: 31.87272
Geodetic CRS: WGS 84

Hide

```
# 读取上海市二手房数据并进行预处理
house_data <- read.csv("上海_geo.csv", fileEncoding = "UTF-8") %>%
  filter(总价.万元. > 0) %>% # 过滤掉总价为0的数据
  mutate(
    配备电梯 = factor(配备电梯, levels = c("有", "无"), labels = c("有电梯", "无电梯")), # 转换电梯配置为因子
    区 = case_when(
      区 == "浦东" ~ "浦东新区", # 统一区的命名, 确保一致性
      TRUE ~ gsub("区", "", 区) # 去除“区”字
    )
  )

# 2. 计算指标 -----
# 计算各区的平均总价
district_price <- house_data %>%
  group_by(区) %>%
  summarise(平均总价 = mean(总价.万元., na.rm = TRUE))

# 计算各区的电梯配置比例
elevator_ratio <- house_data %>%
  count(区, 配备电梯) %>%
  group_by(区) %>%
  mutate(比例 = n / sum(n)) %>%
  ungroup()

# 3. 数据合并 -----
# 将计算出的指标合并到上海市的行政区数据中
sh_shp <- sh_shp %>%
  left_join(district_price, by = c("district" = "区")) %>% # 合并平均总价
  left_join(elevator_ratio, by = c("district" = "区")) # 合并电梯配置比例

# 4. 获取每个区的中心点坐标 -----
# 计算每个行政区的质心坐标
centers <- sh_shp %>%
  st_centroid() %>% # 获取每个几何体的中心点
  st_coordinates() %>% # 获取坐标
  as.data.frame() %>% # 转换为数据框
  rename(center_x = X, center_y = Y) # 重命名坐标列
```

警告: st_centroid assumes attributes are constant over geometries

Hide

```

# 5. 合并质心坐标和原始数据 -----
# 将质心坐标与行政区数据合并
sh_shp_with_centers <- sh_shp %>%
  st_drop_geometry() %>% # 去除几何信息，只保留属性数据
  bind_cols(centers) # 合并质心坐标

# 6. 可视化 -----
ggplot() +
  # 总价热力图层
  geom_sf(data = sh_shp, aes(fill = 平均总价), color = "white", size = 0.2) +
  scale_fill_gradientn(
    name = "总价（万元）",
    colors = c("#E6F5D0", "#B3E2AB", "#66C2A5"),
    na.value = "grey90"
  ) +

  # 电梯配置比例饼图层
  ggnewscale::new_scale_fill() +
  geom_arc_bar(
    data = sh_shp_with_centers %>% drop_na(配备电梯), # 仅使用含电梯配置数据
    aes(x0 = center_x,
        y0 = center_y,
        r0 = 0,
        r = 0.03, # 调整饼图的半径大小
        amount = 比例,
        fill = 配备电梯),
    stat = "pie",
    color = "white",
    size = 0.1,
    inherit.aes = FALSE
  ) +
  scale_fill_manual(
    name = "电梯配置",
    values = c("有电梯" = "#66C2A5", "无电梯" = "#FC8D62"),
    na.value = "grey50"
  ) +

  # 地图元素
  annotation_scale() + # 添加比例尺
  annotation_north_arrow(
    location = "tr", # 设置北箭头的位置
    style = north_arrow_minimal() # 设置北箭头样式
  ) +
  labs(title = "上海市二手房总价与电梯配置分析") + # 设置标题
  theme_void() + # 使用空白主题
  theme(
    legend.position = "right", # 设置图例位置
    plot.title = element_text(hjust = 0.5, face = "bold", size = 14), # 设置标题样式
    legend.text = element_text(size = 9) # 设置图例文本样式
  )

```

上海市二手房总价与电梯配置分析



Hide

NA

NA

上海各区二手房单价及装修情况分析

Hide

```
library(sf)
library(ggplot2)
library(dplyr)
library(ggforce)
library(tidyr)
library(ggspatial)
library(ggnewscale)

# 1. 读取地图和房屋数据 (请替换实际shp文件路径)
sh_shp <- st_read("上海市.shp") %>%
  st_make_valid() %>%
  st_simplify(dTolerance = 0.01) %>%
  mutate(district = gsub("区|市辖区", "", name)) # 假设地理数据包含NAME字段
```

Reading layer `上海市' from data source `D:\data_analysis\R语言可视化\上海市.shp' using driver `ESRI Shapefile'

Simple feature collection with 16 features and 9 fields

Geometry type: MULTIPOLYGON

Dimension: XY

Bounding box: xmin: 120.8568 ymin: 30.67559 xmax: 122.2471 ymax: 31.87272

Geodetic CRS: WGS 84

Hide

```
house_data <- read.csv("上海_geo.csv", fileEncoding = "UTF-8")
colnames(house_data) <- gsub(" ", "", colnames(house_data)) # 去除列名中的空格
colnames(house_data) <- trimws(colnames(house_data)) # 去除列名两端空格

# 再次检查列名
cat("列名检查：\n")
```

列名检查：

Hide

```
print(colnames(house_data))
```

```
[1] "小区名称"      "房屋朝向"      "装修情况"      "供暖方式"      "户型结构"      "建筑
类型"      "建筑结构"      "梯户比例"
[9] "配备电梯"      "挂牌时间"      "上次交易"      "房屋年限"      "抵押信息"      "交易
权属"      "房屋用途"      "产权所属"
[17] "关注人数"      "发布日期"      "VR看装修"      "近地铁"      "随时看房"      "详情
页"      "楼层高低"      "楼层总高度"
[25] "区"      "街道"      "环线位置"      "居住空间"      "功能间"      "总
价.万元."      "单价.元.平."      "房屋面积.平米."
[33] "地址"      "lat"      "lon"
```

Hide

```
house_data <- house_data %>%
  filter(总价.万元. > 0) %>%
  mutate(
    区 = case_when(
      区 == "浦东" ~ "浦东新区", # 处理浦东新区特殊名称
      TRUE ~ gsub("区", "", 区)
    )
  )

# 2. 计算各区指标
district_avg <- house_data %>%
  group_by(区) %>%
  summarise(平均单价 = mean(单价.元.平., na.rm = TRUE))

decoration_ratio <- house_data %>%
  count(区, 装修情况) %>%
  group_by(区) %>%
  mutate(比例 = n / sum(n)) %>%
  ungroup()

# 3. 合并地理数据
sh_shp <- sh_shp %>%
  left_join(district_avg, by = c("district" = "区")) %>%
  left_join(decoration_ratio, by = c("district" = "区"))

# 4. 获取中心点坐标
centers <- sh_shp %>%
  st_centroid() %>%
  st_coordinates() %>%
  as.data.frame() %>%
  rename(center_x = X, center_y = Y)
```

警告: st_centroid assumes attributes are constant over geometries

[Hide](#)

```
# 5. 装修情况点图数据（取各区主要装修类型）
top_decoration <- house_data %>%
  group_by(区, 装修情况) %>%
  summarise(n = n(), .groups = "drop") %>%
  group_by(区) %>%
  slice_max(n, n = 1) %>%
  rename(district = 区)

decoration_points <- sh_shp %>%
  st_drop_geometry() %>%
  select(district) %>%
  bind_cols(centers) %>%
  left_join(top_decoration, by = "district") %>%
  drop_na(装修情况)

# 6. 可视化
ggplot() +
  geom_sf(data = sh_shp, aes(fill = 平均单价), color = "white", size = 0.2) +
  scale_fill_gradientn(
    name = "均价（元/平）",
    colors = c("#E6F5D0", "#B3E2AB", "#66C2A5"),
    na.value = "grey90"
  ) +
  ggnewscale::new_scale_color() +
  geom_point(
    data = decoration_points,
    aes(x = center_x, y = center_y, color = 装修情况),
    size = 3.5, alpha = 0.8
  ) +
  scale_color_manual(
    name = "主要装修情况",
    values = c("精装" = "#66C2A5", "简装" = "#FC8D62", "毛坯" = "#8DA0CB"),
    na.value = "grey50"
  ) +
  annotation_scale() +
  annotation_north_arrow(
    location = "tr",
    style = north_arrow_minimal()
  ) +
  labs(title = "上海市各区二手房单价与装修情况分析") +
  theme_void() +
  theme(
    legend.position = "right",
    plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
    legend.text = element_text(size = 9)
  )
```

上海市各区二手房单价与装修情况分析



广州各区二手房价格及电梯配置分析

Hide

```
library(sf)
library(ggplot2)
library(dplyr)
library(ggforce)
library(tidyr)
library(ggspatial)
library(ggnewscale)

# 1. 数据准备 -----
# 读取广州市的行政区边界数据并进行预处理
gz_shp <- st_read("广州市.shp") %>%
  st_make_valid() %>%
  st_simplify(dTolerance = 0.01) %>%
  mutate(district = gsub("区|市辖区", "", name))
```

Reading layer `广州市` from data source `D:\data_analysis\R语言可视化\广州市.shp` using driver `ESRI Shapefile`
Simple feature collection with 11 features and 9 fields
Geometry type: POLYGON
Dimension: XY
Bounding box: xmin: 112.9585 ymin: 22.51436 xmax: 114.06 ymax: 23.93292
Geodetic CRS: WGS 84

Hide

```
# 读取广州二手房数据并进行预处理
house_data <- read.csv("广州_geo.csv", fileEncoding = "UTF-8") %>%
  filter(总价.万元. > 0) %>%
  mutate(
    配备电梯 = factor(配备电梯, levels = c("有", "无"), labels = c("有电梯", "无电梯")),
    区 = case_when(
      区 == "荔湾" ~ "荔湾",
      TRUE ~ gsub("区", "", 区)
    )
  )

# 2. 计算指标 -----
# 计算各区的平均总价
district_price <- house_data %>%
  group_by(区) %>%
  summarise(平均总价 = mean(总价.万元., na.rm = TRUE))

# 计算各区的电梯配置比例
elevator_ratio <- house_data %>%
  count(区, 配备电梯) %>%
  group_by(区) %>%
  mutate(比例 = n / sum(n)) %>%
  ungroup()

# 3. 数据合并 -----
# 将计算出的指标合并到广州市行政区数据中
gz_shp <- gz_shp %>%
  left_join(district_price, by = c("district" = "区")) %>%
  left_join(elevator_ratio, by = c("district" = "区"))

# 4. 获取每个区的中心点坐标 -----
centers <- gz_shp %>%
  st_centroid() %>%
  st_coordinates() %>%
  as.data.frame() %>%
  rename(center_x = X, center_y = Y)
```

警告: st_centroid assumes attributes are constant over geometries

[Hide](#)

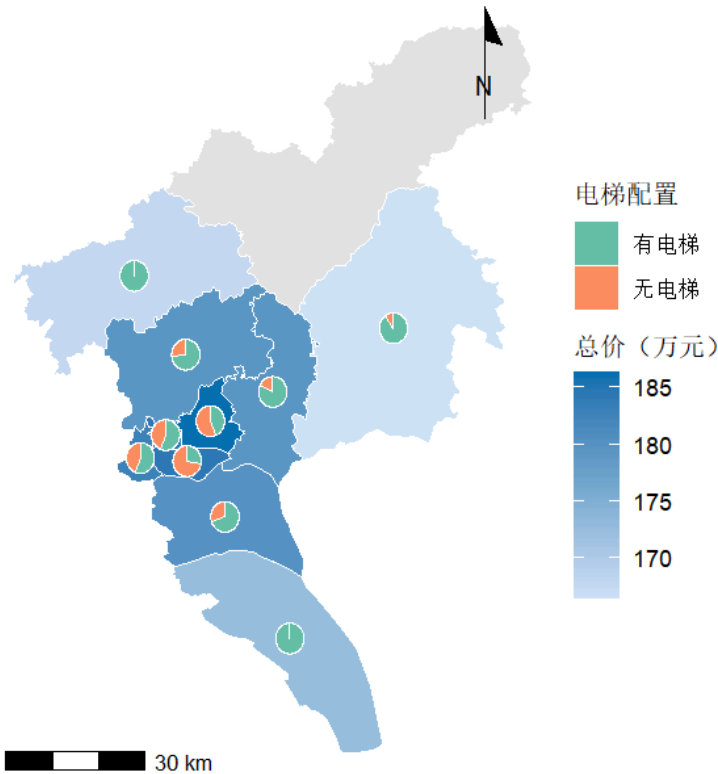

```
# 5. 合并质心坐标和原始数据 -----
gz_shp_with_centers <- gz_shp %>%
  st_drop_geometry() %>%
  bind_cols(centers)

# 6. 可视化 -----
ggplot() +
  # 总价热力图层
  geom_sf(data = gz_shp, aes(fill = 平均总价), color = "white", size = 0.2) +
  scale_fill_gradientn(
    name = "总价（万元）",
    colors = c("#D0E1F9", "#74A9CF", "#0570B0"),
    na.value = "grey90"
  ) +

  # 电梯配置比例饼图层
  ggnewscale::new_scale_fill() +
  geom_arc_bar(
    data = gz_shp_with_centers %>% drop_na(配备电梯),
    aes(x0 = center_x,
        y0 = center_y,
        r0 = 0,
        r = 0.03,
        amount = 比例,
        fill = 配备电梯),
    stat = "pie",
    color = "white",
    size = 0.1,
    inherit.aes = FALSE
  ) +
  scale_fill_manual(
    name = "电梯配置",
    values = c("有电梯" = "#66C2A5", "无电梯" = "#FC8D62"),
    na.value = "grey50"
  ) +

  # 地图元素
  annotation_scale() +
  annotation_north_arrow(
    location = "tr",
    style = north_arrow_minimal()
  ) +
  labs(title = "广州市二手房总价与电梯配置分析") +
  theme_void() +
  theme(
    legend.position = "right",
    plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
    legend.text = element_text(size = 9)
  )
```

广州市二手房总价与电梯配置分析



广州各区二手房单价及装修情况分析

Hide

```
library(sf)
library(ggplot2)
library(dplyr)
library(ggforce)
library(tidyr)
library(ggspatial)
library(ggnewscale)

# 1. 读取广州市地图和二手房数据（请替换为你自己的文件路径）
gz_shp <- st_read("广州市.shp") %>%
  st_make_valid() %>%
  st_simplify(dTolerance = 0.01) %>%
  mutate(district = gsub("区|市辖区", "", name))
```

Reading layer `广州市` from data source `D:\data_analysis\R语言可视化\广州市.shp` using driver `ESRI Shapefile`
Simple feature collection with 11 features and 9 fields
Geometry type: POLYGON
Dimension: XY
Bounding box: xmin: 112.9585 ymin: 22.51436 xmax: 114.06 ymax: 23.93292
Geodetic CRS: WGS 84

Hide

```
house_data <- read.csv("广州_geo.csv", fileEncoding = "UTF-8")
colnames(house_data) <- gsub(" ", "", colnames(house_data)) # 去除列名中空格
colnames(house_data) <- trimws(colnames(house_data))        # 去除首尾空格

cat("列名检查: \n")
```

列名检查:

[Hide](#)

```
print(colnames(house_data))
```

```
[1] "小区名称"      "房屋朝向"      "装修情况"      "供暖方式"      "户型结构"      "建筑
类型"      "建筑结构"      "梯户比例"
[9] "配备电梯"      "挂牌时间"      "上次交易"      "房屋年限"      "抵押信息"      "交易
权属"      "房屋用途"      "产权所属"
[17] "关注人数"      "发布日期"      "VR看装修"      "近地铁"      "随时看房"      "详情
页"      "楼层高低"      "楼层总高度"
[25] "区"      "街道"      "环线位置"      "居住空间"      "功能间"      "总
价.万元."      "单价.元.平."      "房屋面积.平米."
[33] "地址"      "lat"      "lon"
```

[Hide](#)

```
# 数据清洗
house_data <- house_data %>%
  filter(总价.万元. > 0, 单价.元.平. > 0) %>%
  mutate(
    区 = gsub("区", "", 区) # 移除“区”字
  )

# 2. 计算平均单价和装修比例
district_avg <- house_data %>%
  group_by(区) %>%
  summarise(平均单价 = mean(单价.元.平., na.rm = TRUE))

decoration_ratio <- house_data %>%
  count(区, 装修情况) %>%
  group_by(区) %>%
  mutate(比例 = n / sum(n)) %>%
  ungroup()

# 3. 合并地理信息
gz_shp <- gz_shp %>%
  left_join(district_avg, by = c("district" = "区")) %>%
  left_join(decoration_ratio, by = c("district" = "区"))

# 4. 计算中心点坐标
centers <- gz_shp %>%
  st_centroid() %>%
  st_coordinates() %>%
  as.data.frame() %>%
  rename(center_x = X, center_y = Y)
```

警告: `st_centroid` assumes attributes are constant over geometries

[Hide](#)

```
# 5. 计算每区主要装修情况（最多的装修类型）
top_decoration <- house_data %>%
  group_by(区, 装修情况) %>%
  summarise(n = n(), .groups = "drop") %>%
  group_by(区) %>%
  slice_max(n, n = 1) %>%
  rename(district = 区)

# 6. 整合用于点图的数据
decoration_points <- gz_shp %>%
  st_drop_geometry() %>%
  select(district) %>%
  bind_cols(centers) %>%
  left_join(top_decoration, by = "district") %>%
  drop_na(装修情况)
```

警告: Detected an unexpected many-to-many relationship between `x` and `y`.

[Hide](#)

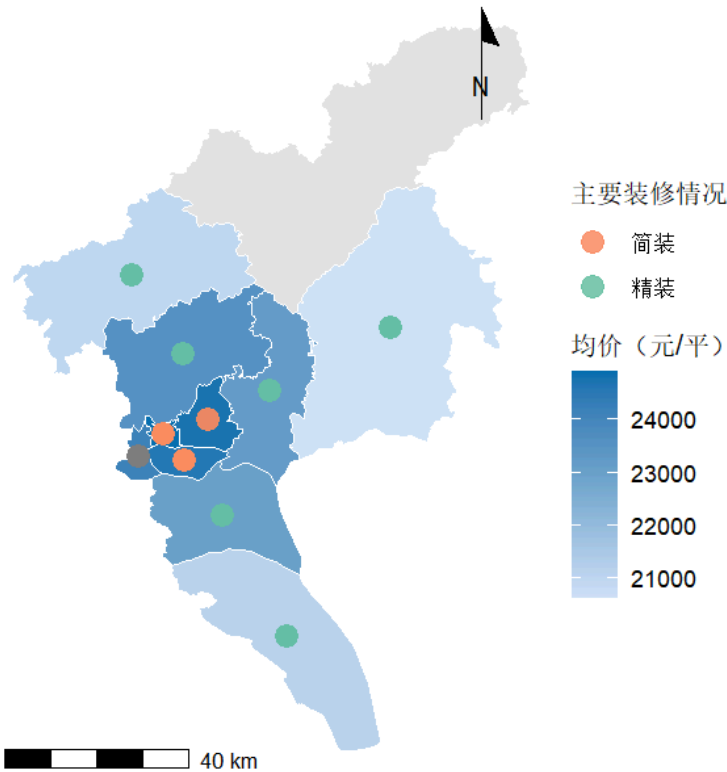
```
# 7. 可视化：蓝色调地图 + 中心点装修分类
ggplot() +
  # 背景地图（平均单价）
  geom_sf(data = gz_shp, aes(fill = 平均单价), color = "white", size = 0.2) +
  scale_fill_gradientn(
    name = "均价（元/平）",
    colors = c("#D0E1F9", "#74A9CF", "#0570B0"),
    na.value = "grey90"
  ) +

  # 新图层：装修情况圆点
  ggnewscale::new_scale_color() +
  geom_point(
    data = decoration_points,
    aes(x = center_x, y = center_y, color = 装修情况),
    size = 4, alpha = 0.85
  ) +
  scale_color_manual(
    name = "主要装修情况",
    values = c("精装" = "#66C2A5", "简装" = "#FC8D62", "毛坯" = "#8DA0CB"),
    na.value = "grey50"
  ) +

  # 地图元素
  annotation_scale(location = "bl", width_hint = 0.4) +
  annotation_north_arrow(location = "tr", style = north_arrow_minimal()) +

  # 标题与主题
  labs(title = "广州市各区二手房单价与装修情况分析") +
  theme_void() +
  theme(
    legend.position = "right",
    plot.title = element_text(hjust = 0.5, face = "bold", size = 15),
    legend.text = element_text(size = 9)
  )
```

广州市各区二手房单价与装修情况分析



深圳各区二手房价格及电梯配置分析

Hide

```
library(sf)
library(ggplot2)
library(dplyr)
library(ggforce)
library(tidyr)
library(ggspatial)
library(ggnewscale)

# 1. 数据准备 -----
# 读取深圳市的行政区边界数据并进行预处理
sz_shp <- st_read("深圳市.shp") %>%
  st_make_valid() %>%
  st_simplify(dTolerance = 0.01) %>%
  mutate(district = gsub("区|市辖区", "", name))
```

Reading layer `深圳市` from data source `D:\data_analysis\R语言可视化\深圳市.shp` using driver `ESRI Shapefile`
Simple feature collection with 9 features and 9 fields
Geometry type: MULTIPOLYGON
Dimension: XY
Bounding box: xmin: 113.7515 ymin: 22.39634 xmax: 114.6285 ymax: 22.86175
Geodetic CRS: WGS 84

Hide

```
# 读取深圳二手房数据并进行预处理
house_data <- read.csv("深圳_geo.csv", fileEncoding = "UTF-8") %>%
  filter(总价.万元. > 0) %>%
  mutate(
    配备电梯 = factor(配备电梯, levels = c("有", "无"), labels = c("有电梯", "无电梯")),
    区 = gsub("区", "", 区)
  )

# 2. 计算指标 -----
# 计算各区的平均总价
district_price <- house_data %>%
  group_by(区) %>%
  summarise(平均总价 = mean(总价.万元., na.rm = TRUE))

# 计算各区的电梯配置比例
elevator_ratio <- house_data %>%
  count(区, 配备电梯) %>%
  group_by(区) %>%
  mutate(比例 = n / sum(n)) %>%
  ungroup()

# 3. 数据合并 -----
sz_shp <- sz_shp %>%
  left_join(district_price, by = c("district" = "区")) %>%
  left_join(elevator_ratio, by = c("district" = "区"))

# 4. 获取每个区的中心点坐标 -----
centers <- sz_shp %>%
  st_centroid() %>%
  st_coordinates() %>%
  as.data.frame() %>%
  rename(center_x = X, center_y = Y)
```

警告: st_centroid assumes attributes are constant over geometries

[Hide](#)

```

# 5. 合并质心坐标和原始数据 -----
sz_shp_with_centers <- sz_shp %>%
  st_drop_geometry() %>%
  bind_cols(centers)

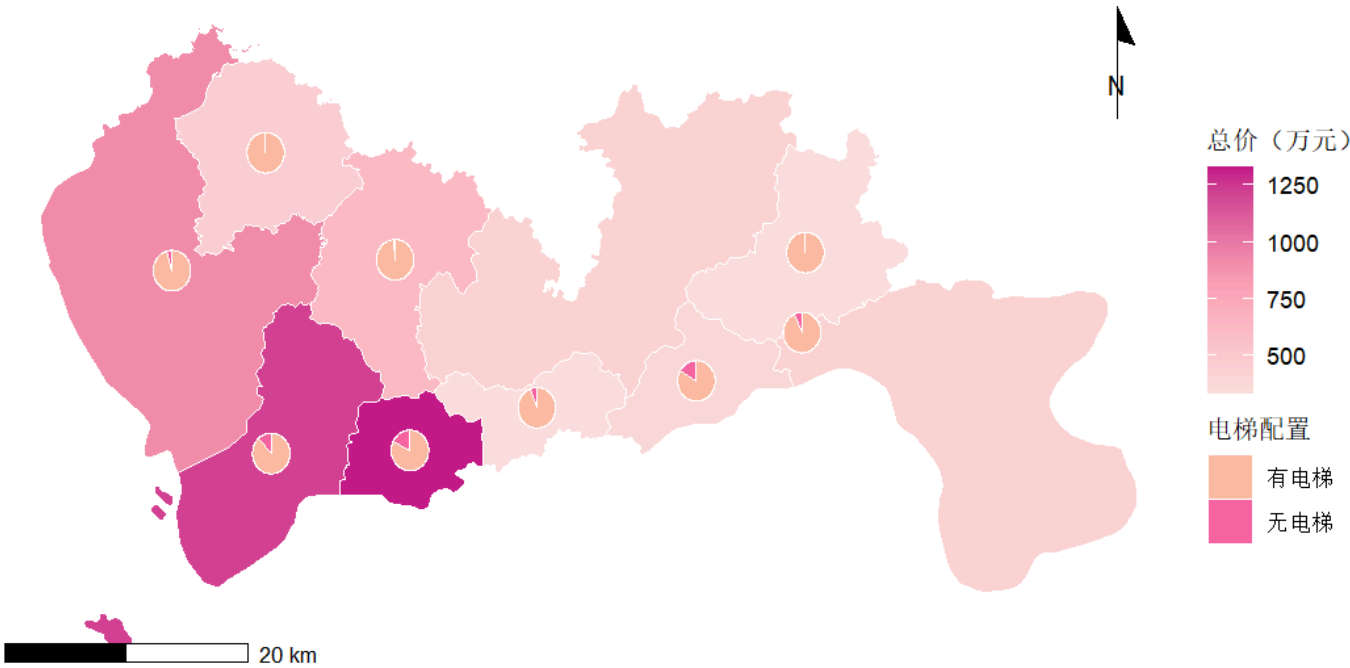
# 6. 可视化 -----
ggplot() +
  # 总价热力图层
  geom_sf(data = sz_shp, aes(fill = 平均总价), color = "white", size = 0.2) +
  scale_fill_gradientn(
    name = "总价（万元）",
    colors = c("#FDE0DD", "#FA9FB5", "#C51B8A"), # 粉红色调
    na.value = "grey90"
  ) +

  # 电梯配置比例饼图层
  ggnewscale::new_scale_fill() +
  geom_arc_bar(
    data = sz_shp_with_centers %>% drop_na(配备电梯),
    aes(x0 = center_x,
        y0 = center_y,
        r0 = 0,
        r = 0.015,
        amount = 比例,
        fill = 配备电梯),
    stat = "pie",
    color = "white",
    size = 0.1,
    inherit.aes = FALSE
  ) +
  scale_fill_manual(
    name = "电梯配置",
    values = c("有电梯" = "#FDBBA2", "无电梯" = "#F768A1"),
    na.value = "grey50"
  ) +

  # 地图元素
  annotation_scale() +
  annotation_north_arrow(
    location = "tr",
    style = north_arrow_minimal()
  ) +
  labs(title = "深圳市二手房总价与电梯配置分析") +
  theme_void() +
  theme(
    legend.position = "right",
    plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
    legend.text = element_text(size = 9)
  )

```


深圳市二手房总价与电梯配置分析



深圳各区二手房单价及装修情况分析

Hide

```
library(sf)
library(ggplot2)
library(dplyr)
library(ggforce)
library(tidyr)
library(ggspatial)
library(ggnewscale)

# 1. 读取深圳市地图和二手房数据（请替换为你自己的文件路径）
sz_shp <- st_read("深圳市.shp") %>%
  st_make_valid() %>%
  st_simplify(dTolerance = 0.01) %>%
  mutate(district = gsub("区|市辖区", "", name))
```

Reading layer `深圳市` from data source `D:\data_analysis\R语言可视化\深圳市.shp` using driver `ESRI Shapefile`
Simple feature collection with 9 features and 9 fields
Geometry type: MULTIPOLYGON
Dimension: XY
Bounding box: xmin: 113.7515 ymin: 22.39634 xmax: 114.6285 ymax: 22.86175
Geodetic CRS: WGS 84

Hide

```
house_data <- read.csv("深圳_geo.csv", fileEncoding = "UTF-8")
colnames(house_data) <- gsub(" ", "", colnames(house_data))
colnames(house_data) <- trimws(colnames(house_data))

cat("列名检查: \n")
```

列名检查:

Hide

```
print(colnames(house_data))
```

```
[1] "小区名称"      "房屋朝向"      "装修情况"      "供暖方式"      "户型结构"      "建筑
类型"      "建筑结构"      "梯户比例"
[9] "配备电梯"      "挂牌时间"      "上次交易"      "房屋年限"      "抵押信息"      "交易
权属"      "房屋用途"      "产权所属"
[17] "关注人数"      "发布日期"      "VR看装修"      "近地铁"      "随时看房"      "详情
页"      "楼层高低"      "楼层总高度"
[25] "区"      "街道"      "环线位置"      "居住空间"      "功能间"      "总
价.万元."      "单价.元.平."      "房屋面积.平米."
[33] "地址"      "lat"      "lon"
```

Hide

```
# 2. 数据清洗
house_data <- house_data %>%
  filter(总价.万元. > 0, 单价.元.平. > 0) %>%
  mutate(区 = gsub("区", "", 区))

# 3. 计算平均单价和装修情况比例
district_avg <- house_data %>%
  group_by(区) %>%
  summarise(平均单价 = mean(单价.元.平., na.rm = TRUE))

decoration_ratio <- house_data %>%
  count(区, 装修情况) %>%
  group_by(区) %>%
  mutate(比例 = n / sum(n)) %>%
  ungroup()

# 4. 合并地理信息
sz_shp <- sz_shp %>%
  left_join(district_avg, by = c("district" = "区")) %>%
  left_join(decoration_ratio, by = c("district" = "区"))

# 5. 计算中心点坐标
centers <- sz_shp %>%
  st_centroid() %>%
  st_coordinates() %>%
  as.data.frame() %>%
  rename(center_x = X, center_y = Y)
```

警告: st_centroid assumes attributes are constant over geometries

```
# 6. 各区最常见装修情况
top_decoration <- house_data %>%
  group_by(区, 装修情况) %>%
  summarise(n = n(), .groups = "drop") %>%
  group_by(区) %>%
  slice_max(n, n = 1) %>%
  rename(district = 区)

# 7. 整合中心点与装修数据
decoration_points <- sz_shp %>%
  st_drop_geometry() %>%
  select(district) %>%
  bind_cols(centers) %>%
  left_join(top_decoration, by = "district") %>%
  drop_na(装修情况)

# 8. 可视化：粉红色调地图 + 中心点装修分类
ggplot() +
  # 背景地图（平均单价）
  geom_sf(data = sz_shp, aes(fill = 平均单价), color = "white", size = 0.2) +
  scale_fill_gradientn(
    name = "均价（元/平）",
    colors = c("#FFE0F0", "#F89FB8", "#DB3A6B"), # 粉红色系
    na.value = "grey90"
  ) +

  # 装修分类圆点图层
  ggnewscale::new_scale_color() +
  geom_point(
    data = decoration_points,
    aes(x = center_x, y = center_y, color = 装修情况),
    size = 4, alpha = 0.85
  ) +
  scale_color_manual(
    name = "主要装修情况",
    values = c("精装" = "#66C2A5", "简装" = "#FC8D62", "毛坯" = "#8DA0CB", "其他" = "#D81B60"),
    na.value = "grey50"
  ) +

  # 地图元素
  annotation_scale(location = "bl", width_hint = 0.4) +
  annotation_north_arrow(location = "tr", style = north_arrow_minimal()) +

  # 标题与主题
  labs(title = "深圳市各区二手房单价与装修情况分析") +
  theme_void() +
  theme(
    legend.position = "right",
    plot.title = element_text(hjust = 0.5, face = "bold", size = 15),
    legend.text = element_text(size = 9)
  )
)
```

深圳市各区二手房单价与装修情况分析

