

```
In [1]: import pandas as pd
df = pd.read_excel(r'D:\bacon seed\scu\Zeronn\college_data.xls')
df.head()
```

Out[1]:

	name	site	title	type	belong	nature	website
0	北京大学	北京	211985	综合	教育部	本科	www.pku.edu.cn
1	中国人民大学	北京	211985	综合	教育部	本科	www.ruc.edu.cn
2	清华大学	北京	211985	工科	教育部	本科	www.tsinghua.edu.cn
3	北京航空航天大学	北京	211985	工科	工业与信息化部	本科	www.buaa.edu.cn
4	北京理工大学	北京	211985	工科	工业与信息化部	本科	www.bit.edu.cn

```
In [2]: df_new = df.drop_duplicates(subset = ['name']) # 剔除重复的数据
df_site = df_new[df_new['site'] != '——']
df_site = df_site[df_site['site'] != '———']
```

```
In [3]: # 高校总数量分析
site_counts = df_site['site'].value_counts() # Series 时序
dict_site = {'name': site_counts.index, 'counts': site_counts.values} # 字典
data = pd.DataFrame(dict_site) # 转换DataFrame格式
```

```
In [4]: # 各城市高校数量
from pyecharts.charts import Bar
from pyecharts import options as opts
from pyecharts.commons.utils import JsCode

bar = Bar()
bar.add_xaxis(data['name'].values.tolist())
bar.add_yaxis("", data['counts'].values.tolist())
bar.set_global_opts(
    axis_opts=opts.AxisOpts(axislabel_opts=opts.LabelOpts(rotate=-15)),
    title_opts=opts.TitleOpts(title="各城市高校数量", subtitle="仅仅统计数量"),
    datazoom_opts=opts.DataZoomOpts(),
)
bar.set_series_opts(itemstyle_opts={
    "normal": {
        "color": JsCode("""new echarts.graphic.LinearGradient(0, 0, 0, 1, [
            offset: 0,
            color: 'rgba(0, 244, 255, 1)'
        ]), {
            offset: 1,
            color: 'rgba(0, 77, 167, 1)'
        }], false)"""),
        "barBorderRadius": [30, 30, 30, 30],
        "shadowColor": 'rgb(0, 160, 221)',
    })
```

Out[4]: <pyecharts.charts.basic_charts.bar.Bar at 0x2831bc14940>

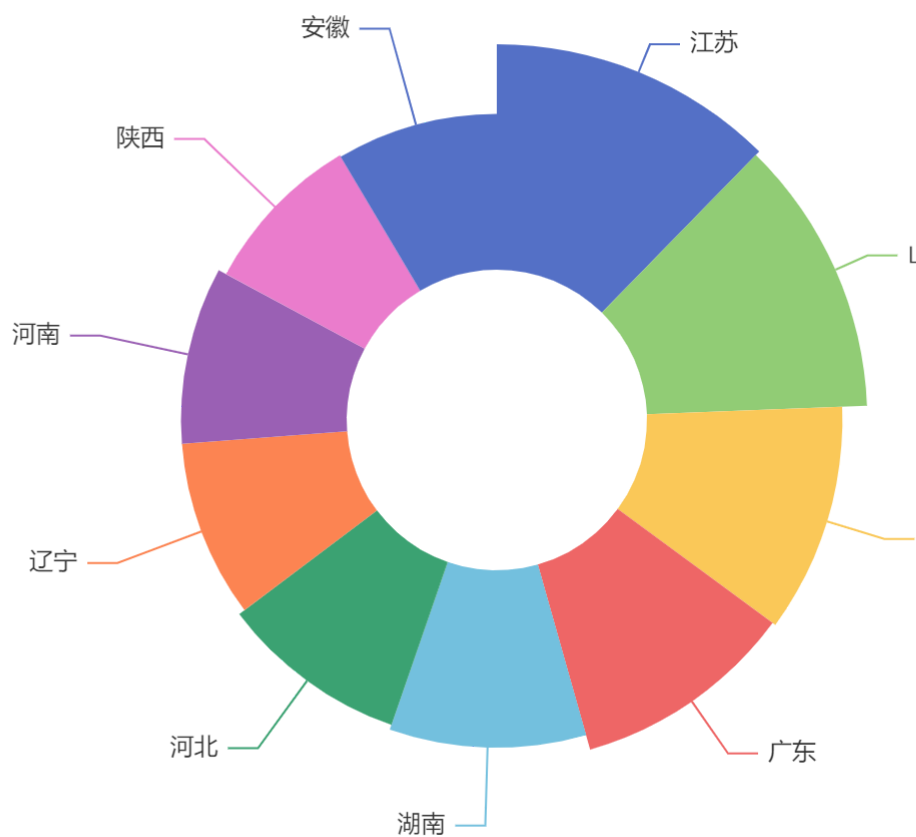
```
In [5]: site_counts
```

```
Out[5]: 江苏      136  
        山东      133  
        湖北      118  
        广东      116  
        湖南      107  
        河北      104  
        辽宁      100  
        河南      100  
        陕西      95  
        安徽      94  
        北京      91  
        浙江      90  
        四川      87  
        江西      83  
        黑龙江    81  
        福建      71  
        山西      68  
        上海      66  
        广西      63  
        天津      60  
        吉林      55  
        云南      50  
        重庆      49  
        贵州      45  
        新疆      38  
        甘肃      38  
        内蒙古    31  
        海南      15  
        宁夏      13  
        青海       9  
        香港       8  
        西藏       5  
        澳门       2  
Name: site, dtype: int64
```

```
In [6]: from pyecharts.charts import Pie

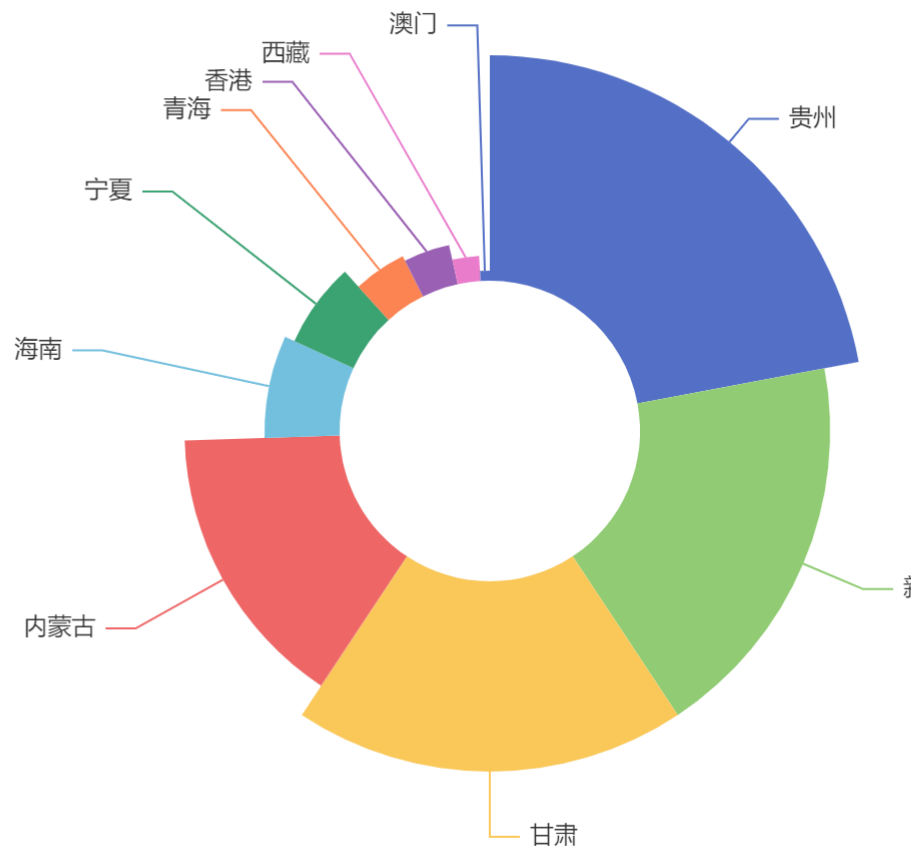
pie = Pie()
pie.add("", [list(z) for z in zip(data['name'].values.tolist()[:10], data['counts'].values.tolist()[:10])],
        radius=["30%", "75%"],
        center=["40%", "50%"],
        rosetype="radius")
pie.set_global_opts(
    title_opts=opts.TitleOpts(title="高校数量前十名"),
    legend_opts=opts.LegendOpts(
        type_="scroll", pos_left="80%", orient="vertical"
    ),
)
pie.render_notebook()
```

Out[6]: **高校数量前十名**



```
In [7]: pie = Pie()
pie.add("", [list(z) for z in zip(data['name'].values.tolist()[-10:], data['counts']
radius=["30%", "75%"],
center=["40%", "50%"],
rosetype="radius")
pie.set_global_opts(
title_opts=opts.TitleOpts(title="高校数量后十名"),
legend_opts=opts.LegendOpts(
type_="scroll", pos_left="80%", orient="vertical"
),
)
pie.render_notebook()
```

Out[7]: **高校数量后十名**

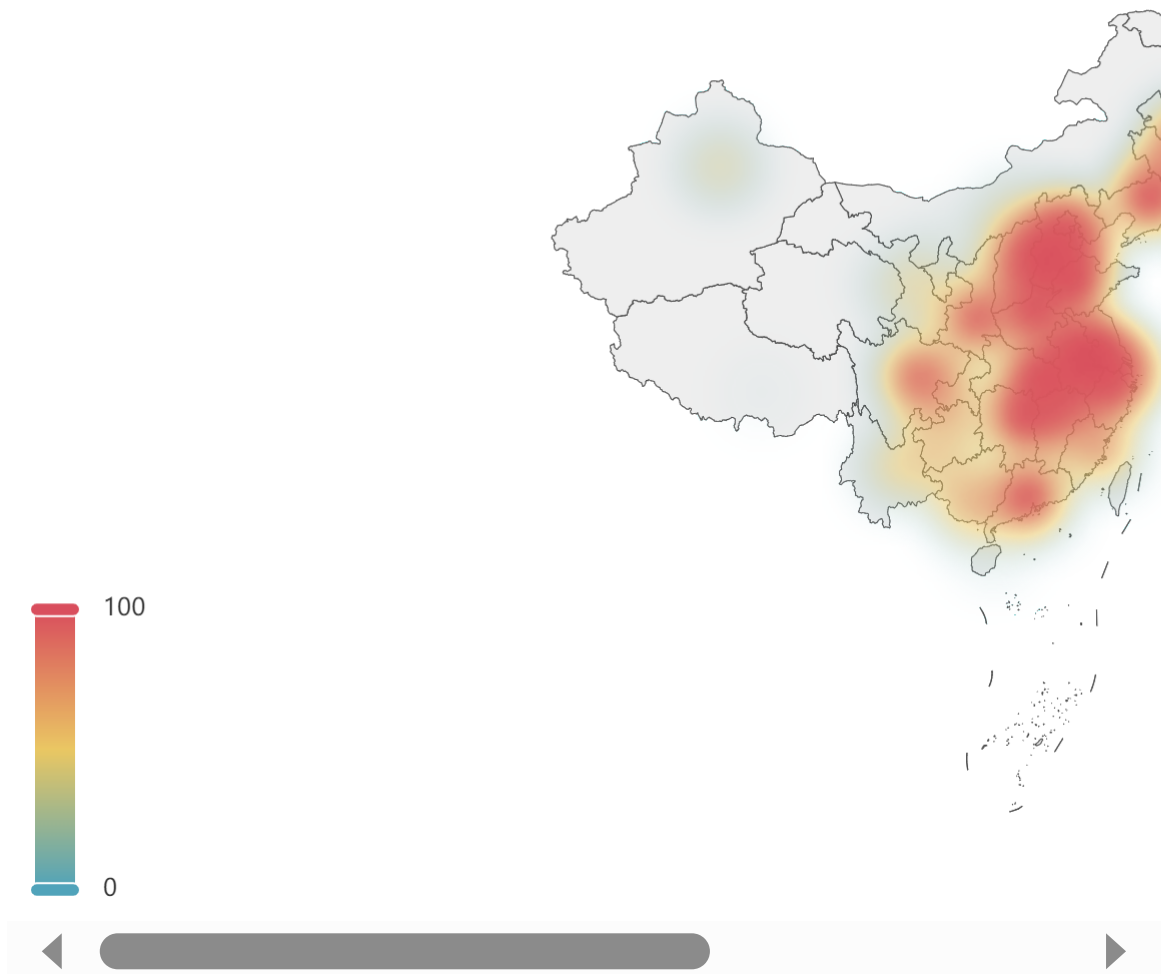


```
In [11]: # 高校分布热力图
from pyecharts.charts import Geo
from pyecharts.globals import ChartType

geo = Geo()
geo.add_schema(maptype="china")
geo.add("高校分布热力图", [list(z) for z in zip(data['name'].values.tolist(), data
        type_=ChartType.HEATMAP)
geo.set_global_opts(
    visualmap_opts=opts.VisualMapOpts(),
    title_opts=opts.TitleOpts(title="全国高校热力图"),
)
geo.render_notebook()
```

Out[11]: **全国高校热力图**

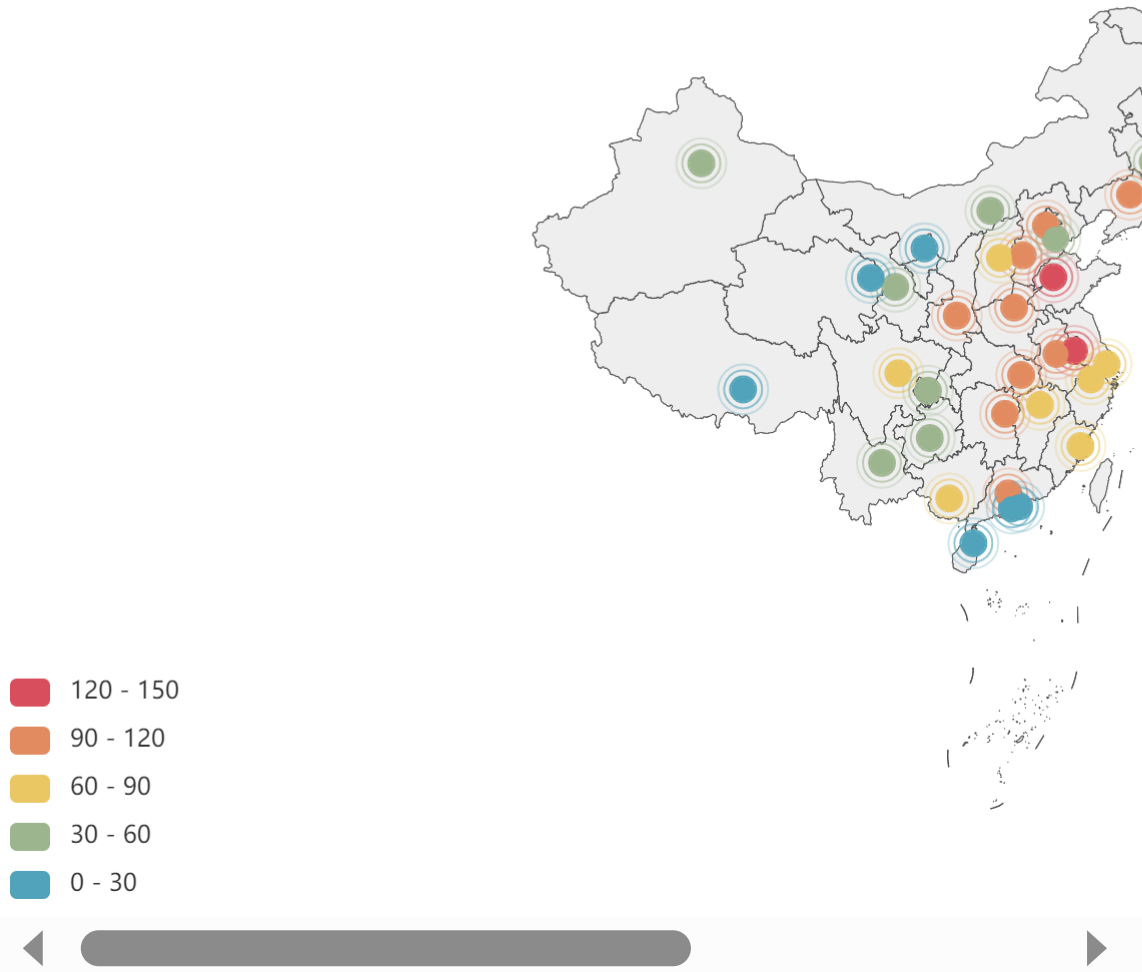
高校分布热力图



```
In [12]: geo = Geo()
geo.add_schema(maptype="china")
geo.add("各地区高校数量段位图", [list(z) for z in zip(data['name'].values.tolist(),
type=ChartType.EFFECT_SCATTER)
geo.set_global_opts(
    visualmap_opts=opts.VisualMapOpts(is_pieewise=True, max_=150),
    title_opts=opts.TitleOpts(title="各地区高校数量段位图"),
)
geo.set_series_opts(label_opts=opts.LabelOpts(is_show=False))
geo.render_notebook()
```

Out[12]: 各地区高校数量段位图

● 各地区高校数量段位图

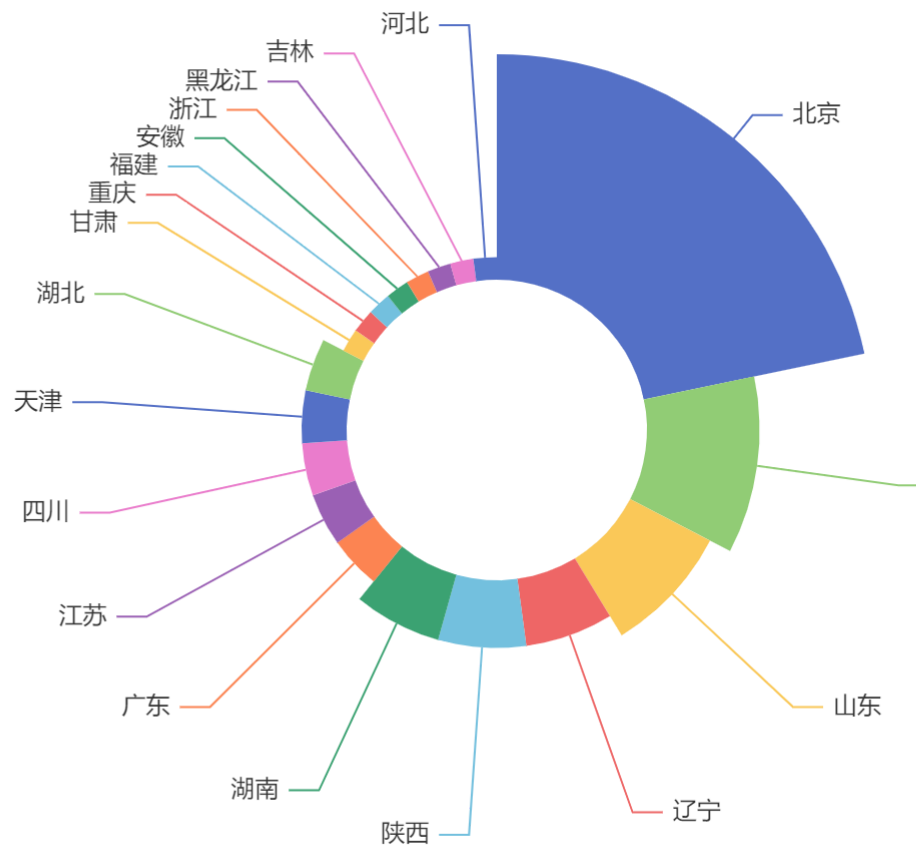


```

In [14]: df_title = df_new[df_new['title'] != '——']
df_985 = df_title[df_title['title'] == 211985]
site_counts_985 = df_985['site'].value_counts()
dict_site_985 = {'name': site_counts_985.index, 'counts': site_counts_985.values}
data_985 = pd.DataFrame(dict_site_985)
# 985高校排行
pie = Pie()
pie.add("", [list(z) for z in zip(data_985['name'].values.tolist(), data_985['counts'].values)],
        radius=["30%", "75%"],
        center=["40%", "50%"],
        rosetype="radius")
pie.set_global_opts(
    title_opts=opts.TitleOpts(title="985高校排行"),
    legend_opts=opts.LegendOpts(
        type_="scroll", pos_left="80%", orient="vertical"
    ),
)
pie.render_notebook()

```

Out[14]: **985高校排行**



```
In [15]: df_211 = df_title[df_title['title'] == 211]
site_counts_211 = df_211['site'].value_counts()
dict_site_211 = {'name': site_counts_211.index, 'counts': site_counts_211.values}
data_211 = pd.DataFrame(dict_site_211)
# 211高校排行
pie = Pie()
pie.add("", [list(z) for z in zip(data_211['name'].values.tolist(), data_211['counts'].values)],
        radius=["30%", "75%"],
        center=["40%", "50%"],
        rosetype="radius")
pie.set_global_opts(
    title_opts=opts.TitleOpts(title="211高校排行"),
    legend_opts=opts.LegendOpts(
        type_="scroll", pos_left="80%", orient="vertical"
    ),
)
pie.render_notebook()
```

Out[15]: **211高校排行**

