

基于 Python 语言，使用 matplotlib、seaborn 绘制图像

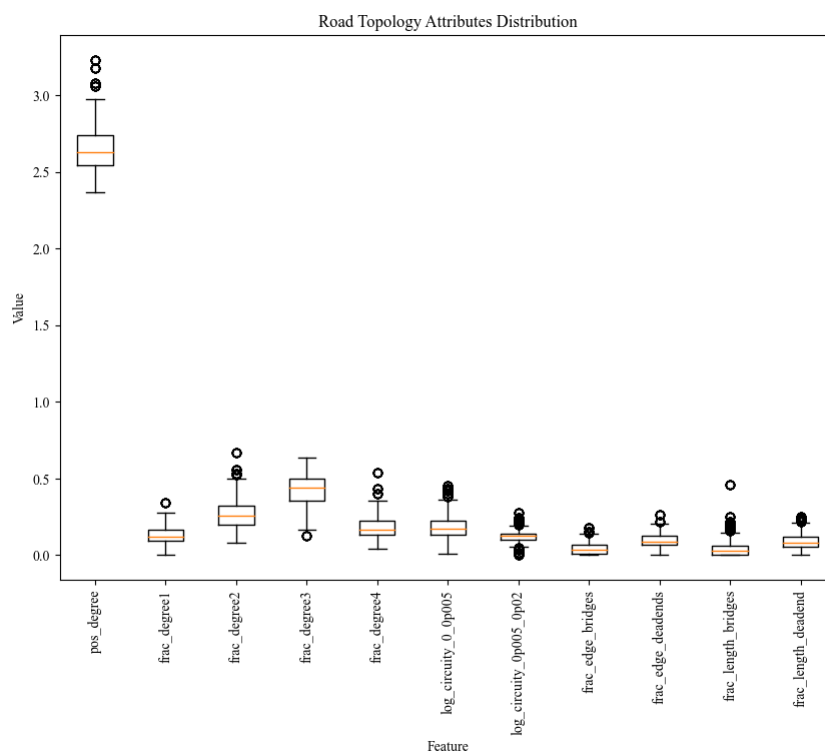
箱图

相关代码：

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv(r"D:\road_economy\data\t_road\road_locs.csv")
features = ['pos_degree', 'frac_degree1', 'frac_degree2',
            "frac_degree3", "frac_degree4",
            "log_circuitry_0_0p005", "log_circuitry_0p005_0p02",
            "frac_edge_bridges",
            "frac_edge_deadends", "frac_length_bridges",
            "frac_length_deadend"]
fig, ax = plt.subplots(figsize=(10, 8))
ax.boxplot(df[features].values, labels=features)
ax.set_xlabel('Feature')
ax.set_ylabel('Value')
ax.set_title('Road Topology Attributes Distribution')
plt.xticks(rotation=90)
plt.subplots_adjust(bottom=0.2)
plt.show()
```

图像效果：



雷达图

相关代码：

```
import csv
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA

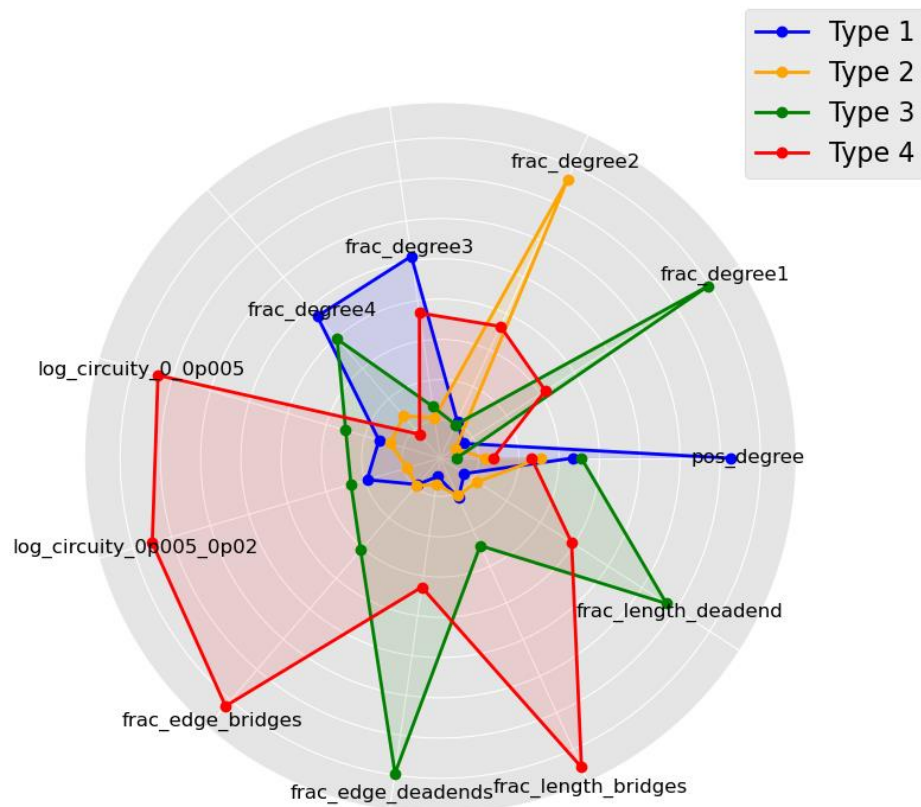
with open("D:\谷歌下载\clusters.csv") as csv_file:
    reader = csv.reader(csv_file)
    data = list(reader)
data_columns = [row[2:14] for row in data[1:]]
data_array = np.array(data_columns).astype(float)
data_mean = np.mean(data_array, axis=0, keepdims=True)
data_std = np.std(data_array, axis=0, keepdims=True)
data = (data_array - data_mean) / data_std
k = 4
kmeanModel = KMeans(n_clusters=k, random_state = 1)
kmeanModel.fit(data)
pca = PCA(n_components=6)
newData = pca.fit_transform(data)
change_order = True
if change_order:
    SDNi = np.zeros(k)
    change_order_mapping = {}
    for i in range(k):
        SDNi[i] = np.mean(newData[kmeanModel.labels_== i][:,0])
    argsorted_SDNi = np.argsort(SDNi)
    for i in range(k):
        change_order_mapping[i] = np.where(argsorted_SDNi==i)[0][0]
    for i in range(len(kmeanModel.labels_)):
        kmeanModel.labels_[i] =
change_order_mapping[kmeanModel.labels_[i]]
cluster_centers_ = np.zeros((k, data.shape[1]))
for i in range(k):
    cluster_centers_[i,:] = np.mean(data[kmeanModel.labels_== i], axis =
0, keepdims=True)
name_list = ['pos_degree', 'frac_degree1', 'frac_degree2',
"frac_degree3", "frac_degree4", "log_circuitry_0_0p005",
"log_circuitry_0p005_0p02", "frac_edge_bridges",
"frac_edge_deadends", "frac_length_bridges",
"frac_length_deadend"]
```

```

num_list = cluster_centers_[0]
plt.style.use('ggplot')
fig = plt.figure(figsize=(9.8, 7))
ax = fig.add_subplot(111, polar=True)
feature = name_list
values = cluster_centers_[0]
N = len(values)-1
angles = np.linspace(0, 2 * np.pi, N, endpoint=False)
angles = np.concatenate((angles, [angles[0]]))
color = ['b', 'orange', 'g', 'r']
for i in range(cluster_centers_.shape[0]):
    values = cluster_centers_[i]
    values = np.concatenate((values, [values[0]]))
    ax.plot(angles, values[0:12], 'o-', linewidth=2, label = 'Type ' +
str(i+1), color = color[i])
    ax.fill(angles, values[0:12], alpha=0.1, color = color[i])
ax.grid(True)
ax.yaxis.set_ticklabels([])
for i, angle in enumerate(angles[:-1]):
    label = feature[i]
    x = angle
    y = np.max(cluster_centers_[i], i) * 1.1
    ax.text(x, y, label, ha='center', va='center', fontsize=12)
plt.legend(loc = 'best', bbox_to_anchor=(0.7, 0.65, 0.5, 0.5), fontsize
= 16)
ax.set_xticks(angles[:-1])
ax.set_xticklabels([])
plt.tight_layout()
plt.show()

```

图像效果：



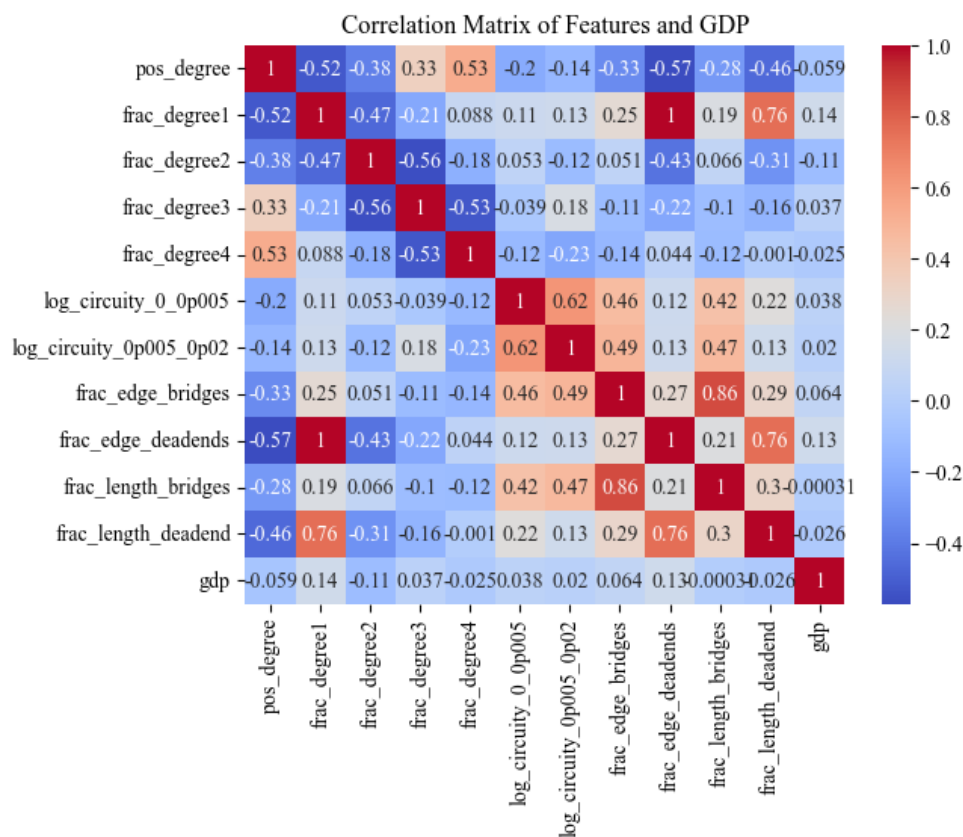
相关矩阵热力图

相关代码：

```
import seaborn as sns
import pandas as pd

data = pd.read_csv("D:\谷歌下载\clusters.csv")
features = ['pos_degree', 'frac_degree1', 'frac_degree2',
            "frac_degree3", "frac_degree4", "log_circuitry_0_0p005",
            "log_circuitry_0p005_0p02", "frac_edge_bridges", "frac_edge_deadends",
            "frac_length_bridges", "frac_length_deadend"]
target = 'gdp'
correlation_matrix = data[features + [target]].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix of Features and GDP')
```

图像效果：



散点图

相关代码：

```
import matplotlib.pyplot as plt
import pandas as pd

data = pd.read_csv("D:\谷歌下载\clusters.csv")
features = ['pos_degree', 'frac_degree1', 'frac_degree2',
            "frac_degree3", "frac_degree4", "log_circuitry_0_0p005",
            "log_circuitry_0p005_0p02", "frac_edge_bridges", "frac_edge_deadends",
            "frac_length_bridges", "frac_length_deadend"]
target = 'gdp'
for feature in features:
    plt.scatter(data[feature], data[target], label=feature)
plt.xlabel('Features')
plt.ylabel('GDP')
plt.title('Scatter plot of Features vs. GDP')
plt.legend()
plt.show()
```

图像效果：

