Chapter 8 - Comparison of taxa and period of building origin

November 10, 2020

[1]:
```
import pandas as pd
import geopandas as gpd
import seaborn as sns
import matplotlib.pyplot as plt
import husl
from legendgram import legendgram
import mapclassify
from matplotlib_scalebar.scalebar import ScaleBar
from matplotlib.colors import ListedColormap
from shapely.geometry import Point

from tqdm import tqdm
```

[2]:
```
clusters = pd.read_csv('/Users/martin/Dropbox/Academia/Data/Geo/Prague/
↪Clustering/complete data/200218_clusters_complete_n20.csv', index_col=0)
```

[3]:
```
years = pd.read_csv('/Users/martin/Dropbox/Academia/Data/Geo/Prague/Validation/
↪origin_year.csv', index_col=0)
```

[4]:
```
years['year'] = pd.Categorical(years['ROK_PUVODNI'],
                               categories=[1840, 1880, 1920, 1950, 1970, 1990,␣
↪2012],
                               ordered=True)
```

[5]:
```
joined = clusters.merge(years[['uID', 'year']], on='uID', how='left')
```

[7]:
```
joined.head(4)
```

[7]:

	uID	cluster	year
0	0	1	1990
1	1	11	1840
2	2	13	1970
3	3	18	1950

[32]:
```
buildings = gpd.read_file('/Users/martin/Dropbox/Academia/Data/Geo/Prague/
↪Clustering/geometry.gpkg', layer='buildings')
```

[33]:
```
buildings = buildings.merge(joined, on='uID', how='left')
```

[46]:
```
buildings.year.unique()
```

[46]:
```
[1990, 1840, 1970, 1950, NaN, 1920, 2012, 1880]
Categories (7, int64): [1840 < 1880 < 1920 < 1950 < 1970 < 1990 < 2012]
```

[ ]:
```
```

[7]:
```
import numpy as np

def show_values_on_bars(axs):
    def _show_on_single_plot(ax):
        for p in ax.patches:
            _x = p.get_x() + p.get_width() / 2
            _y = p.get_y() + p.get_height() + 0.02
            value = '{:.2f}'.format(p.get_height())
            ax.text(_x, _y, value, ha="center")

    if isinstance(axs, np.ndarray):
        for idx, ax in np.ndenumerate(axs):
            _show_on_single_plot(ax)
    else:
        _show_on_single_plot(axs)

pal = [husl.husl_to_hex(*color) for color in colors]
```

[ ]:
```
# historical core
data = joined.loc[joined['cluster'].isin([11])]['year'].
 ↪value_counts(sort=False, normalize=True)

sns.set(context="paper", style="ticks", rc={'patch.force_edgecolor': False})
fig, ax = plt.subplots(figsize=(10, 5))
sns.barplot(ax=ax, x=data.index, y=data, order=data.index, palette=pal)
sns.despine(offset=10)
plt.ylabel('frequency')
plt.xlabel('historical period')
plt.ylim(0, 1)
show_values_on_bars(ax)
```

[ ]:
```
fig, ax = plt.subplots(2, 2, figsize=(10, 7))

data = joined.loc[joined['cluster'].isin([11])]['year'].
 ↪value_counts(sort=False, normalize=True)
sns.barplot(ax=ax[0, 0], x=data.index, y=data, order=data.index, palette=pal)
sns.despine(offset=10)
ax[0,0].set_ylabel('frequency')
```

```
ax[0,0].set_xlabel('historical period')
ax[0,0].set_title('cluster 11')
ax[0,0].set_ylim(0, 1)
show_values_on_bars(ax[0, 0])

data = joined.loc[joined['cluster'].isin([5])]['year'].value_counts(sort=False,␣
 ↪normalize=True)
sns.barplot(ax=ax[0, 1], x=data.index, y=data, order=data.index, palette=pal)
sns.despine(offset=10)
ax[0,1].set_ylabel('frequency')
ax[0,1].set_xlabel('historical period')
ax[0,1].set_title('cluster 5')
ax[0,1].set_ylim(0, 1)
show_values_on_bars(ax[0, 1])

data = joined.loc[joined['cluster'].isin([12])]['year'].
 ↪value_counts(sort=False, normalize=True)
sns.barplot(ax=ax[1, 0], x=data.index, y=data, order=data.index, palette=pal)
sns.despine(offset=10)
ax[1,0].set_ylabel('frequency')
ax[1,0].set_xlabel('historical period')
ax[1,0].set_title('cluster 12')
ax[1,0].set_ylim(0, 1)
show_values_on_bars(ax[1, 0])

data = joined.loc[joined['cluster'].isin([13])]['year'].
 ↪value_counts(sort=False, normalize=True)
sns.barplot(ax=ax[1, 1], x=data.index, y=data, order=data.index, palette=pal)
sns.despine(offset=10)
ax[1,1].set_ylabel('frequency')
ax[1,1].set_xlabel('historical period')
ax[1,1].set_title('cluster 13')
ax[1,1].set_ylim(0, 1)
show_values_on_bars(ax[1, 1])

plt.tight_layout()
plt.savefig('figures/PRG_cluster_origin_subplot.pdf')
```

[ ]:
```
fig, ax = plt.subplots(1, 2, figsize=(10, 3.5))

data = joined.loc[joined['cluster'].isin([11, 15, 5])]['year'].
 ↪value_counts(sort=False, normalize=True)
sns.barplot(ax=ax[0], x=data.index, y=data, order=data.index, palette=pal)
sns.despine(offset=10)
ax[0].set_ylabel('frequency')
ax[0].set_xlabel('historical period')
ax[0].set_title('compact core')
```

```
ax[0].set_ylim(0, 1)
show_values_on_bars(ax[0])

data = joined.loc[joined['cluster'].isin([1, 19])]['year'].
 ↪value_counts(sort=False, normalize=True)
sns.barplot(ax=ax[1], x=data.index, y=data, order=data.index, palette=pal)
sns.despine(offset=10)
ax[1].set_ylabel('frequency')
ax[1].set_xlabel('historical period')
ax[1].set_title('large scale industry')
ax[1].set_ylim(0, 1)
show_values_on_bars(ax[1])
plt.tight_layout()

plt.savefig('figures/PRG_branch_origin_subplot.pdf')
```

[43]:
```
# save all clusters
for cl in range(20):
    data = joined.loc[joined['cluster'].isin([cl])]['year'].
 ↪value_counts(sort=False, normalize=True)

    sns.set(context="paper", style="ticks", rc={'patch.force_edgecolor': False})
    fig, ax = plt.subplots(figsize=(10, 5))
    sns.barplot(ax=ax, x=data.index, y=data, order=data.index, palette=pal)
    sns.despine(offset=10)
    plt.ylabel('frequency')
    plt.xlabel('historical period')
    plt.ylim(0, 1)
    show_values_on_bars(ax)
    for ext in ['pdf', 'png']:
        plt.savefig('figures/PRG_cluster_' + str(cl) + '_origin.' + ext,␣
 ↪bbox_inches='tight')
    plt.close()
```

[ ]:
```
# blocks core
data = joined.loc[joined['cluster'].isin([11, 15, 5, 10])]['year'].
 ↪value_counts(sort=False, normalize=True)

sns.set(context="paper", style="ticks", rc={'patch.force_edgecolor': False})
fig, ax = plt.subplots(figsize=(10, 5))
sns.barplot(ax=ax, x=data.index, y=data, order=data.index, palette=pal)
sns.despine(offset=10)
plt.ylabel('frequency')
plt.xlabel('historical period')
plt.ylim(0, 1)
show_values_on_bars(ax)
for ext in ['pdf', 'png']:
```

```
    plt.savefig('figures/PRG_compact_' + str(cl) + '_origin.' + ext,␣
 ↪bbox_inches='tight')
```

[ ]:
```
# modern core
data = joined.loc[joined['cluster'].isin([12, 14])]['year'].
 ↪value_counts(sort=False, normalize=True)

sns.set(context="paper", style="ticks", rc={'patch.force_edgecolor': False})
fig, ax = plt.subplots(figsize=(10, 5))
sns.barplot(ax=ax, x=data.index, y=data, order=data.index, palette=pal)
sns.despine(offset=10)
plt.ylabel('frequency')
plt.xlabel('historical period')
plt.ylim(0, 1)
show_values_on_bars(ax)
for ext in ['pdf', 'png']:
    plt.savefig('figures/PRG_modern_' + str(cl) + '_origin.' + ext,␣
 ↪bbox_inches='tight')
```

years and their clusters

[82]:
```
colors = [(257, 71, 27), (98, 93, 78), (14, 79, 58), (26, 0, 50), (75, 90, 85),␣
 ↪(347, 72, 60), (246, 79, 60)]
cols = []
for col in colors:
    pal = sns.light_palette(col, input="husl", n_colors=5)
    for rgb in pal[1:]:
        cols.append(rgb)

symbology = {0: cols[10],
             1: cols[14],
             2: cols[20],
             3: cols[9],
             4: cols[22],
             5: cols[1],
             6: cols[18],
             7: cols[23],
             8: cols[11],
             9: cols[21],
             10: cols[0],
             11: cols[3],
             12: cols[7],
             13: cols[22],
             14: cols[6],
             15: cols[2],
             16: cols[19],
             17: cols[23],
```

```
            18: cols[17],
            19: cols[15]}
```

[ ]:
```
for p in joined.year.cat.categories:

    data = joined[joined.year == p].cluster.value_counts(normalize=True)

    fig, ax = plt.subplots(figsize=(10, 5))
    sns.barplot(ax=ax, x=data.index, y=data, palette=symbology)
    sns.despine(offset=10)
    plt.ylim(0, 1)
    plt.ylabel('frequency')
    plt.xlabel('cluster')
    show_values_on_bars(ax)
    for ext in ['pdf', 'png']:
        plt.savefig('figures/PRG_period_' + str(cl) + '.' + ext,␣
↪bbox_inches='tight')
```

These have to be interpeted, but some taxa show remarkable link between year of construction and urban pattern. YAY!

[8]:
```
import scipy.stats as ss
import numpy as np

def cramers_v(x, y):
    confusion_matrix = pd.crosstab(x,y)
    chi2 = ss.chi2_contingency(confusion_matrix)[0]
    n = confusion_matrix.sum().sum()
    phi2 = chi2/n
    r,k = confusion_matrix.shape
    phi2corr = max(0, phi2-((k-1)*(r-1))/(n-1))
    rcorr = r-((r-1)**2)/(n-1)
    kcorr = k-((k-1)**2)/(n-1)
    return np.sqrt(phi2corr/min((kcorr-1),(rcorr-1)))
```

[9]:
```
cramers_v(joined.cluster, joined.year)
```

[9]:
```
0.35805157150989764
```

The resulting value of 0.5 indicates moderate relationship between clustering and historical origin.

Chi-square test of independence of variables in a contingency table

[10]:
```
confusion_matrix = pd.crosstab(joined.cluster, joined.year)
chi, p, dof, exp = ss.chi2_contingency(confusion_matrix)
```

[11]:
```
p
```

[11]: 0.0

[12]:
```
dof
```

[12]: 114

[13]:
```
chi
```

[13]: 106700.50861461915

[15]:
```
joined.cluster.size
```

[15]: 140315

p-value is < 0.001

there is a significant dependency between variables