Németh Péter
2016.07.21.

Megoldott feladatok:

-kedd 6. + ábra(cellák élkapcsolata)
-dashboard-hoz feladatok módosítása, interaktivitásának növelése

Feladat:

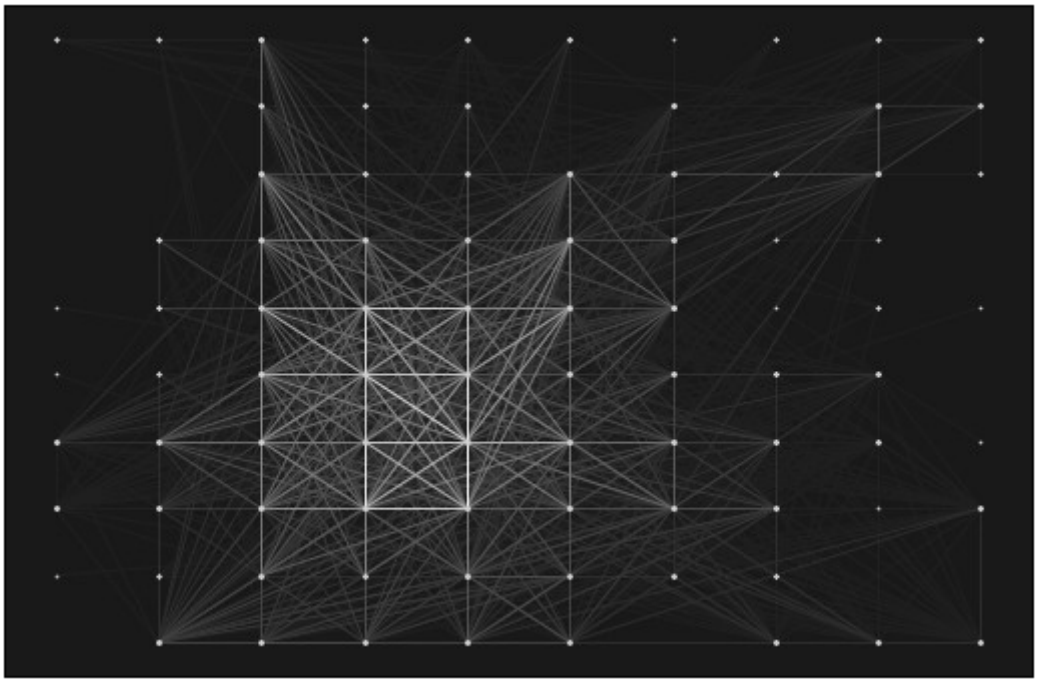
Két cella között annyi élt húzunk, ahány közös device-hoz tartozó esemény szerepel bennük. Melyik három cellapár esetén van a legtöbb él a pár két cellája között?

Kód:

```
import re
df_cj=df_cj.sort_values(by='cnt',ascending=False)
lon_min, lon_max = 116, 117 #Beijing
lat_min, lat_max = 39.75, 40.25
m = Basemap(projection='merc',
        llcrnrlat=lat_min,
        urcrnrlat=lat_max,
        llcrnrlon=lon_min,
        urcrnrlon=lon_max,
        lat_ts=35,
        resolution='c')
fig = plt.figure(101, figsize=(12,6))
m.fillcontinents(color='#191919',lake_color='#000000') # dark grey land, black lakes
m.drawmapboundary(fill_color='#000000')                # black background
m.drawcountries(linewidth=0.1, color="w")
def szinero(n):
    kar=re.sub('0x','',hex(n))
    if len(kar)==1:
        kar='0'+kar
    return ('#'+kar*3)
#print(szinero(34))
harom = []
i=0
for t in df_cj.itertuples():
    harom.append((t.sz1,t.sz2))
    i+=1
    if (i==3):
        break


df_cj=df_cj.sort_values(by='cnt',ascending=True)
cnt_mx = df_cj.cnt.max()
for t in df_cj.itertuples():
    mxy = m([szt[t.sz1][0],szt[t.sz2][0]],[szt[t.sz1][1],szt[t.sz2][1]])
    #print(mxy)
    if(t.cnt>0):
        #print(szinero(int(math.floor(t.cnt/float(cnt_mx)*255))))
        m.scatter(mxy[0], mxy[1], s=3, c='lightgrey', lw=0, alpha=1, zorder=5)
        m.plot(mxy[0], mxy[1], '-', label=x,
color=szinero(int(math.floor(t.cnt**.7/float(cnt_mx**.7)*243+12))))
plt.show()
print('A három legtöbb élszámú cellakapcsolat cellapárjai:')
for t in harom:
    print t
```

Kimenet:

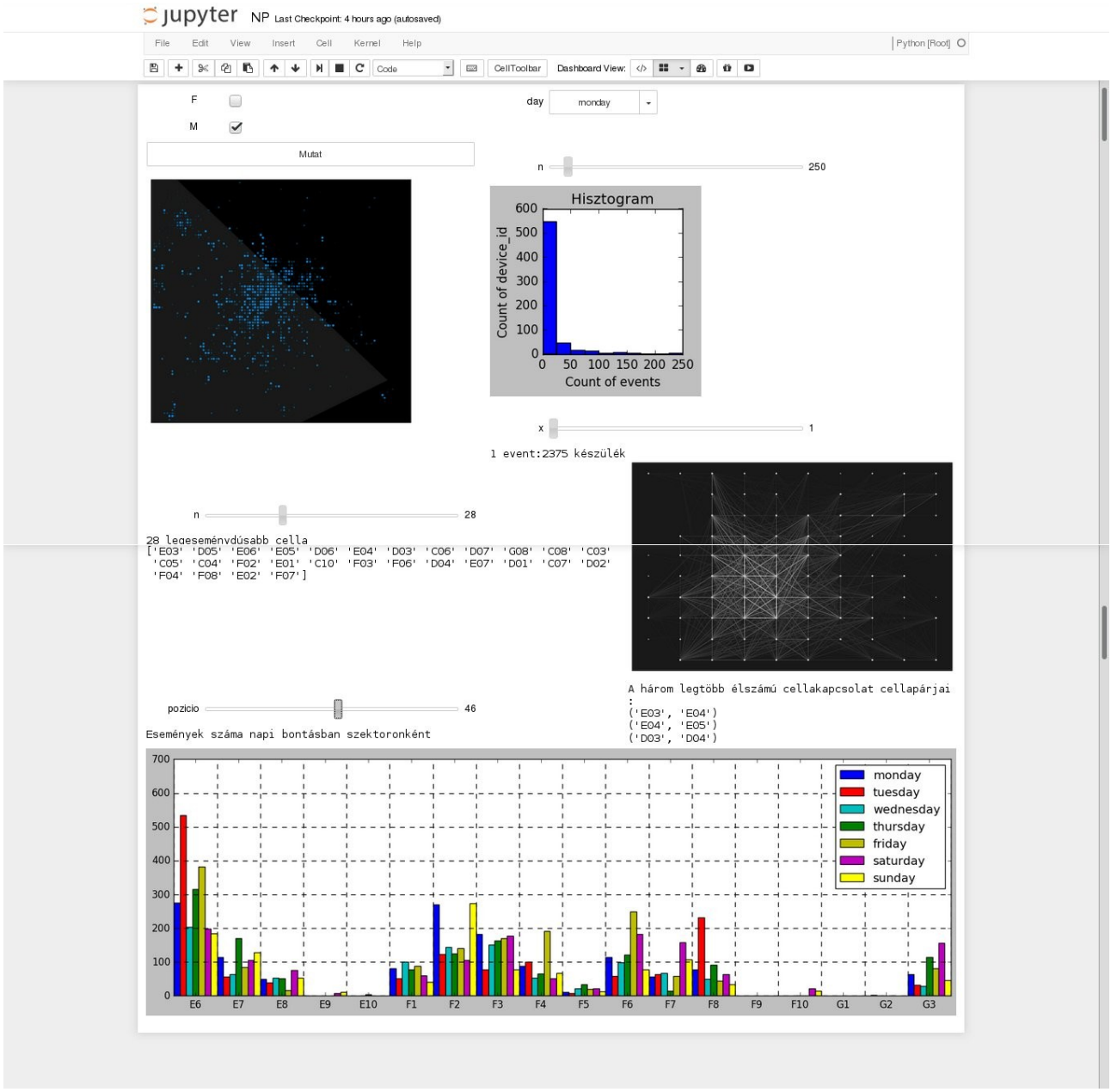


```
A három legtöbb élszámú cellakapcsolat cellapárjai:
('E03', 'E04')
('E04', 'E05')
('D03', 'D04')
```

Feladat:

Korábbi feladatok dashboardra való továbbfejlesztése

Megoldás:
(ld. következő oldal)



Kódok:

```
...
#Események száma napi bontásban
df_tmp =
dfp[['hetnapja','szektor','event_id']].groupby(['hetnapja','szektor'],as_index=False).count().rename(columns={'eve
nt_id':'event_cnt'})
df_tmp['s'] =df_tmp.apply(lambda r:hetnapja[r['hetnapja']],axis=1)
df_tmp = df_tmp.sort_values(by=['s','szektor'])
del df_tmp['s']
X=[]
Y=[]
for d in df_tmp[['hetnapja']].drop_duplicates().values:
    s= d[0]
    i = hetnapja[s]
    Y.append([])
    X.append([])
    Y[i]=Y[i]+list(df_tmp.query('hetnapja==@s').event_cnt.values)
    def szektorertek(inp):
        return (ord(inp[0])-ord('A'))*10+int(inp[1:])
    for v in df_tmp.query('hetnapja==@s').szektor.values:
        X[i].append(szektorertek(v))

def f(pozicio):
    plt.rcdefaults()
    fig, ax = plt.subplots()
    for xc in np.arange(100+1+1):
        plt.axvline(x=xc, color='k', linestyle='--')
    for xc in np.arange(600+1,step=100):
```

```
        plt.axhline(y=xc, color='k', linestyle='--')
    #plt.xlabel('szektor')
    #plt.ylabel('event_cnt')


    width = 1/float(len(Y))
    colors=['b','r','c','g','y','m','yellow']
    sg=[]
    for i in range(0, len(Y)):
        sg.append([])
        sg[i]=ax.bar([e+width*i for e in X[i]],Y[i] , width, color=colors[i])
    ax.legend((sg[0], sg[1], sg[2], sg[3], sg[4], sg[5], sg[6]), ('monday',
'tuesday','wednesday','thursday','friday','saturday','sunday'))
        #lusta voltam kikeresni, inkább beírtam
    ax.set_xticks([i+0.5 for i in np.arange(start=1,stop=100+1)])
    l= []# labels, az üres sztring pedig nulla
    for i in range(1,101):
        c=i
        if (c%10==0):
            c-=10
        kar=chr(ord('A')+int(math.floor(c/10)))
        d=c%10
        if(d==0):
            d=10
        l.append(kar+str(d))
    ax.set_xticklabels(l)
    ax.set_xlim(pozicio,pozicio+18) #102)#(50,70) # utóbbi dokumentációhoz részletkép (hogy kiférjen A4-re)
    fig.set_size_inches(17, 5)#(10,10) #
    print('Események száma napi bontásban szektoronként')
    #print(df_tmp)
    fig.canvas.draw()
interact(f,pozicio=(1-3,101-12,8))
...

...
#Hisztogram
def h(n):
    plt.rcdefaults()
    plt.figure(3, figsize=(2.5,2.5))
    plt.hist(df_cnt2.frequency.values,range=[0, n], bins=10, facecolor='blue')
    plt.title("Hisztogram")
    plt.xlabel("Count of events")
    plt.ylabel("Count of device_id")
    plt.show()
mn=df_cnt2.event_cnt.min()
mx=df_cnt2.event_cnt.max()
interact(h, n=(math.floor((mn+9)/10)*10,math.floor(mx/10)*10,10))
...

...
#Legeseménydúsabb cellák
df_dv_sz_tmp =
dfp[['szektor','device_id']].drop_duplicates().groupby(['device_id'],as_index=False).count().rename(columns={'szek
tor':'szektor_cnt'}).sort_values(by='szektor_cnt',ascending=False)
df_dv_sz= df_dv_sz_tmp.groupby(['szektor_cnt'],as_index=False).count().rename(columns={'device_id':'devices'})

print('Hány device szerepel a mérés ideje alatt n db cellában is?')
display(df_dv_sz)
plt.xlabel('szektor_cnt')
plt.ylabel('devices')
plt.plot(df_dv_sz.szektor_cnt, df_dv_sz.devices)
plt.show()
df_es_sz=dfp[['szektor','event_id']].groupby(['szektor'],as_index=False).count().rename(columns={'event_id':'event
_cnt'})
df_es_sz = df_es_sz.sort_values(by=['event_cnt'] ,ascending=False)

print('Cellák események száma szerint rendezve')
for line in df_es_sz.values:
    print line
vals = df_es_sz.szektor.values
def f(n):
    print('{} legeseménydúsabb cella'.format(n))
    print(vals[:n])
interact(f,n=(1,len(vals),1)).
...

...
#Shanghai férfiak-nők
# Sample it down to only the Shanghai region
lon_min, lon_max = 121.0, 122.0     #121.522179
lat_min, lat_max = 30.8, 31.6 #31.267401

from IPython.display import display, clear_output

# Load the train data and join on the events
df_train = pd.read_csv("./mobil/gender_age_train.csv", dtype={'device_id': np.str})

df_plot = pd.merge(df_train, df_events_shanghai, on="device_id", how="inner")
```

```
df_m = df_plot[df_plot["gender"]=="M"]
df_f = df_plot[df_plot["gender"]=="F"]



cb_f=widgets.Checkbox(description = 'F', value=True, width=90)
cb_m=widgets.Checkbox(description = 'M', value=True, width=90)
btn = widgets.Button(description="Mutat")
display(cb_f)
display(cb_m)
display(btn)


def on_btn_clicked(b):
    # Male/female plot

    m = Basemap(projection='merc',
     llcrnrlat=lat_min,
     urcrnrlat=lat_max,
     llcrnrlon=lon_min,
     urcrnrlon=lon_max,
     lat_ts=35,
     resolution='c')
    fig = plt.figure(101, figsize=(12,6))
    m.fillcontinents(color='#191919',lake_color='#000000') # dark grey land, black lakes
    m.drawmapboundary(fill_color='#000000')                # black background
    m.drawcountries(linewidth=0.1, color="w")              # thin white line for country borders
    mxy = m(df_m["longitude"].tolist(), df_m["latitude"].tolist())
    if(cb_m.value):
        m.scatter(mxy[0], mxy[1], s=5, c="#1292db", lw=0, alpha=0.1, zorder=5)
    mxy = m(df_f["longitude"].tolist(), df_f["latitude"].tolist())
    if(cb_f.value):
        m.scatter(mxy[0], mxy[1], s=5, c="#fd3096", lw=0, alpha=0.1, zorder=5)
    print(mxy[0][2])
    clear_output(True)

btn.on_click(on_btn_clicked)
...
```