Megoldott feladatok:

- Kedd 1.-7. feladatok befejezése. A 8.-at PA oldotta meg.
- Sikerült feltelepítenem a dashboard-ot mindhárom gépre (sajnos egymásra továbbra sem látnak át localhost-on)

Feladat:

1. Készítsünk táblázatot, histogramot, hogy a train adatokon mennyi device esetén szerepel csak 1 event, mennyi device esetén 2, stb. Készítsünk egy pont vagy vonal grafikont, valamint a lehetséges értékeket 10 egyenlő kosárba (bin) beosztva oszlopgrafikon is készüljön.

2. Osszuk fel Pekinget (Pekinget nagyjából lefedő téglalapot) egyforma cellákra, összesen 100-ra. (A fedő téglalap oldalait osszuk ehhez 10 egyenlő részre.) A bal alsó cella legyen az A1, mellette B1, stb., a jobb felső cella a J10. Mennyi event tartozik egy cellába? Mennyi a Pekingre, valamint a cellákra számított fiúk-lányok aránya? Mely cellákban tér el jelentősen ez az arány a pekingi aránytól?

3. A 2-es feladat celláit vizsgáljuk meg korosztályok szerint. Mennyi Peking esetében a koreloszlás, és mennyi a cellákban, melyik tér el jelentősen a pekingi eloszlástól?

4. Mennyi különböző device szerepel mindegyik cellában, mennyi device csak 99-ben, stb. mennyi device csak 1-ben. Készítsünk táblázatot és grafikon.

5. Mennyi esemény szerepel a cellákban, melyik ezek közül a három legnagyobb értékű.

6. 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?

7. Hány esemény történik naponta, óránként, hétfőn, kedden, stb, vasárnap Pekingben összesen, és cellánként? Táblázatban, grafikonon ábrázoljuk.

Kód:

```
import numpy as np
import pandas as pd
import time, math, datetime
from IPython.display import display
import itertools

start_cpu = time.clock()
start_real = time.time()

df_gat= pd.read_csv('mobil/gender_age_train.csv')
df_ev = pd.read_csv('mobil/events.csv')
df = pd.merge(df_gat,df_ev)

# 1. feladat
df_cnt1 =
df[['device_id','event_id']].groupby('device_id',as_index=False).count().rename(columns=
{'event_id':'event_cnt'})
df_cnt2 =
df_cnt1.groupby('event_cnt',as_index=False).count().rename(columns={'device_id':'frequen
cy'})

print('Hány device szerepel a mérés ideje alatt n event?')
print(df_cnt2)
```

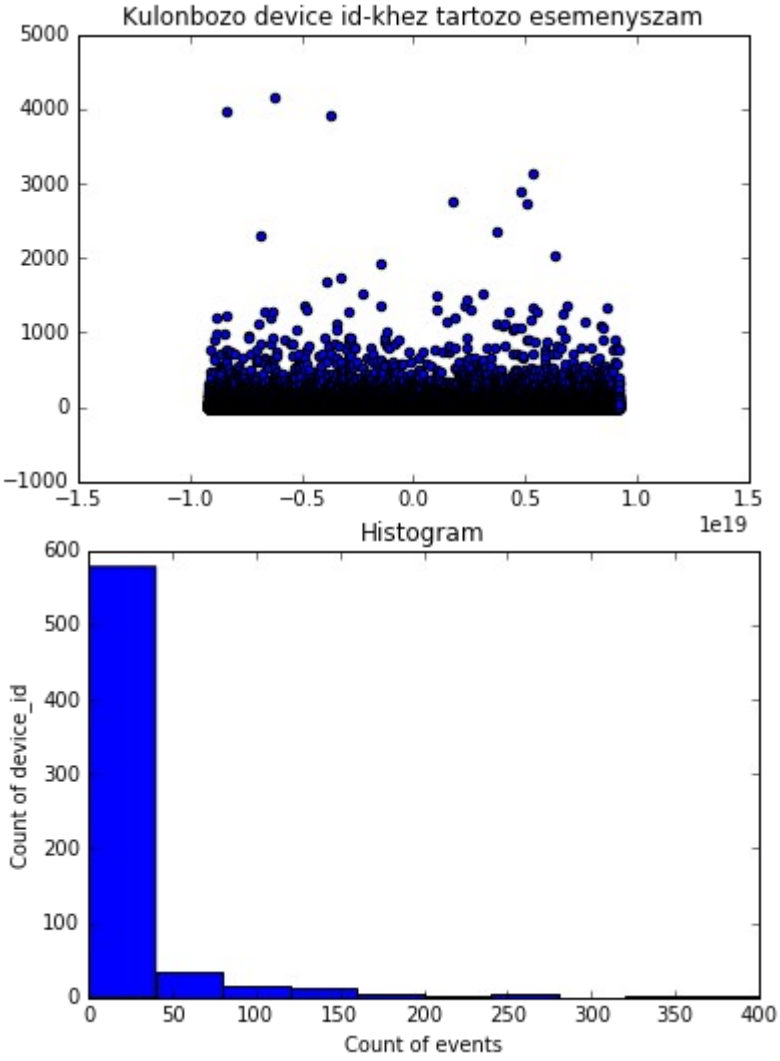
```
%matplotlib inline
import matplotlib.pyplot as plt

print('Hisztogram:')
plt.hist(df_cnt2.frequency.values,range=[0, 400], bins=10, facecolor='blue')
plt.title("Histogram")
plt.xlabel("Count of events")
plt.ylabel("Count of device_id")
plt.show()
plt.title("Kulonbozo device id-khez tartozo esemenyszam")
plt.scatter(df_cnt1.device_id, df_cnt1.event_cnt)
plt.show()

#2. feladat
dfp = df.query('longitude>=116 and longitude<117 and latitude>=39.75 and
latitude<40.25').copy()
def szektor(r):
    abc = ['A','B','C','D','E','F','G','H','I','J']
    num = ['01','02','03','04','05','06','07','08','09','10']
    lon = abc[int(math.floor((r['longitude']-116)*10))]
    lat = num[int(math.floor((r['latitude']-39.75)*20))]
    return lon+lat
dfp['szektor'] = dfp.apply(lambda r : szektor(r),axis=1)

print('Eventek cellánként')
for line in
dfp[['szektor','event_id']].groupby('szektor',as_index=False).count().rename(columns={'e
vent_id':'event_cnt'}).values:
    print line

p_ar_nok=dfp.query('gender=="F"').gender.count()/float(df_gat.shape[0])

print('Peking női eventek aránya')
print (p_ar_nok)

df_ev_nok
=dfp[['szektor','gender']].query('gender=="F"').groupby('szektor',as_index=False).count(
).rename(columns={'gender':'noi_event'})
df_ev_sz
=dfp[['szektor','gender']].groupby('szektor',as_index=False).count().rename(columns={'ge
nder':'osszes_event'})
df_ar_nok = pd.merge(df_ev_nok,df_ev_sz)
df_ar_nok['arany'] = df_ar_nok.noi_event/df_ar_nok.osszes_event
del df_ar_nok['osszes_event']
del df_ar_nok['noi_event']

print('Peking női eventek aránya szektoronként')
for line in df_ar_nok.values:
    print line

df_tmp=df_ar_nok.arany - p_ar_nok
df_ar_nok['elteres'] = df_tmp.abs()
df_ar_nok =df_ar_nok.sort_values(by='elteres',ascending=False)

print('Nők arányának szektoronkénti eltérése a pekingi átlagtól')
for line in df_ar_nok[['szektor', 'elteres']].values:
    print line

p_ag =dfp.age.mean()

print('Pekingi átlagéletkor')
print(p_ag)

df_ag_sz
=dfp[['szektor','age']].groupby('szektor',as_index=False).mean().rename(columns={'age':'
age_avg'})

print('Szektoronkénti átlagéletkor')
```

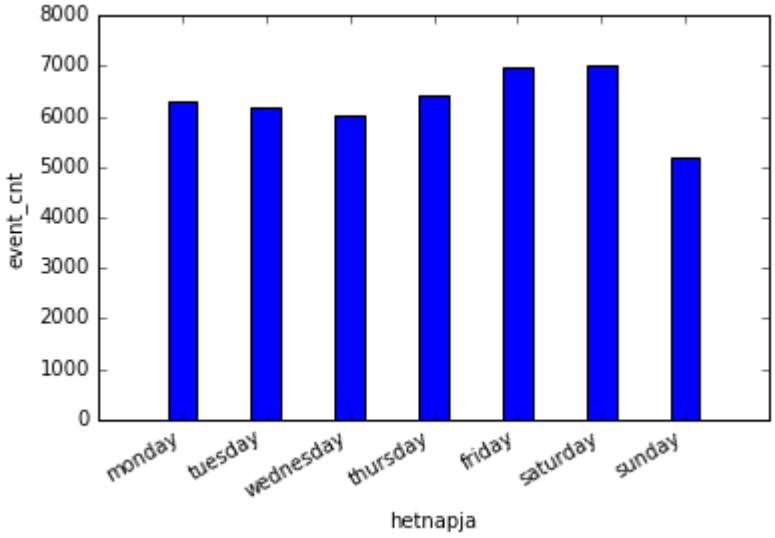
```
for line in df_ag_sz.values:
    print line

df_tmp=df_ag_sz.age_avg - p_ag
df_ag_sz['elteres'] = df_tmp #.abs()
df_ag_sz['abs_elt'] = df_ag_sz['elteres'].abs()
df_ag_sz =df_ag_sz.sort_values(by='abs_elt',ascending=False)
del df_ag_sz['abs_elt']

print('Átlagéletkor szektoronkénti eltérése a pekingi átlagtól')
for line in df_ag_sz[['szektor', 'elteres']].values:
    print line

df_dv_sz_tmp =
dfp[['szektor','device_id']].drop_duplicates().groupby(['device_id'],as_index=False).cou
nt().rename(columns={'szektor':'szektor_cnt'}).sort_values(by='szektor_cnt',ascending=Fa
lse)
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

print('Három legeseménydúsabb cella')
print(df_es_sz.szektor.values[0:3])

hetnapja={'monday':0,'tuesday':1,'wednesday':2,'thursday':3,'friday':4,'saturday':5,'sun
day':6}
hetnapja.update({v: k for k, v in hetnapja.items()})
dfp['hetnapja']=dfp.apply(lambda r:
hetnapja[datetime.datetime.strptime(r['timestamp'],'%Y-%m-%d %H:%M:
%S').weekday()],axis=1)

df_tmp =
dfp[['hetnapja','event_id']].groupby('hetnapja',as_index=False).count().rename(columns={
'event_id':'event_cnt'}).copy()
df_tmp['s'] =df_tmp.apply(lambda r:hetnapja[r['hetnapja']],axis=1)
df_tmp = df_tmp.sort_values(by='s')
del df_tmp['s']

print('Események száma napi bontásban')
display(df_tmp)

X=[]
for v in df_tmp.hetnapja.values:
    X.append(hetnapja[v])
fig, ax = plt.subplots()
plt.xlabel('hetnapja')
plt.ylabel('event_cnt')
plt.bar(X, df_tmp.event_cnt,align='center',width=0.35)
plt.xticks(X,df_tmp.hetnapja)
fig.autofmt_xdate()
plt.show()

df_tmp =
dfp[['hetnapja','szektor','event_id']].groupby(['hetnapja','szektor'],as_index=False).co
unt().rename(columns={'event_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=[]
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')
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))

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(0, 102)#(50,70) # utóbbi dokumentációhoz részletkép (hogy kiférjen A4-re)
fig.set_size_inches(50, 10)#(10,10) #

print('Események száma napi bontásban szektoronként')
#print(df_tmp)
plt.show()

print('6. feladat')
sz=list(np.unique(dfp.szektor.values))
cj=[]
while(0<len(sz)):
    v=sz[0]
    sz.remove(v)
    cj=cj+list(itertools.product([v],sz))
df_cj = pd.DataFrame.from_records(cj,columns=['sz1','sz2'])

def kozos(r):
    sz1=r['sz1']
    sz2=r['sz2']
    result = pd.Series(list(set(dfp.query("szektor==@sz2").device_id) &
```

```
set(dfp.query("szektor==@sz1").device_id)))
    #Series(list(set(s1) & set(s2)))
    return len(result)
df_cj['cnt']=df_cj.apply(lambda r:kozos(r),axis=1)
df_cj=df_cj.sort_values(by='cnt',ascending=False)
print(df_cj)

print('CPU idő: %f sec, Valós idő: %f sec' % (time.clock() - start_cpu, time.time() -
start_real))
```

Kimenet (csak a grafikonok):





Végül egy részlet a széles szektoronkénti és naponkénti bontású eventekből (az eredeti túl széles):