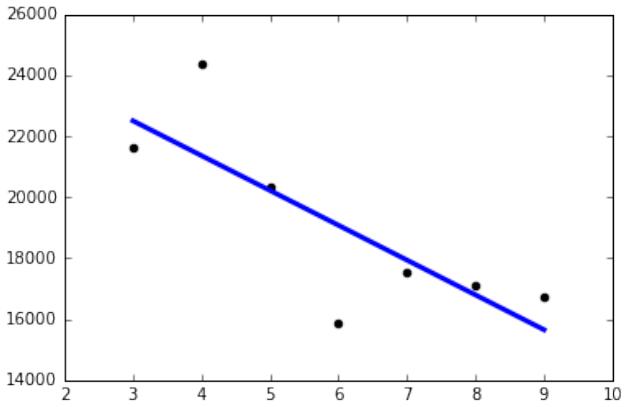
Németh Péter
WN7U8T
2016.07.12

Megoldott feladatok:

- mai 3 feladat
- plusz egy kis javítás a lineáris regresszió megoldásnál:

```
    # JAVITAS
    df=df.groupby('Semana',as_index=False).sum()
    print(df)
    #
```



Így már megfelelő eredményt ad.

Feladat:

Vizsgáljuk meg a train halmazon, vagy egy részén, hogy a termék ára befolyásolja-e az Adjusted Demand értéket. A termék árát megkapjuk, ha az eladott értéket (Venta_hoy) osztjuk az eladott darabszámmal (Venta_uni_hoy). Ábrázoljuk az (ár, Adjusted Demand) pontpárokat! Megfigyelhető-e a grafikonon valamilyen összefüggés, például, hogy a drágább termékekből kevesebbet hoznak vissza?

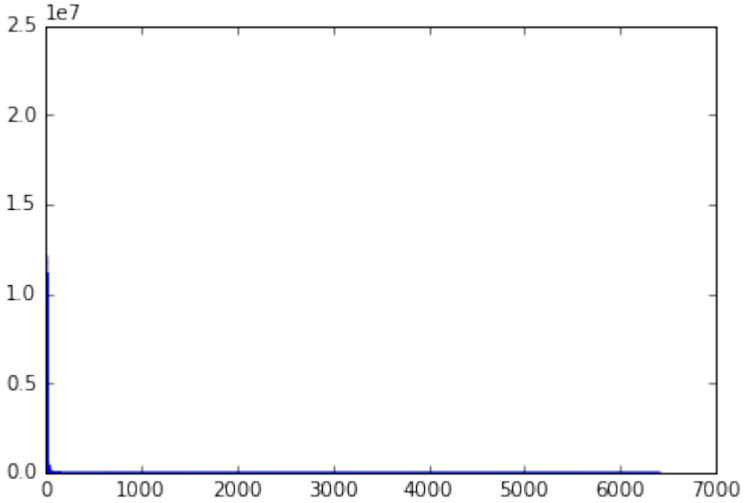
Kód:

```
from __future__ import division
import pandas as pd
import numpy as np
import re, time
import random
start_cpu = time.clock()
start_real = time.time()

%matplotlib inline
import matplotlib.pyplot as plt
df =
pd.read_csv('./input/train.csv',usecols=['Producto_ID','Venta_uni_hoy','Venta_ho
y','Demanda_uni_equil']).\
    groupby('Producto_ID',as_index=False).sum()
#df['ar']=df.apply(lambda row:row['Venta_hoy']/row['Venta_uni_hoy'],axis=1)
#print(df)
df.loc[:,'ar'] = df.loc[:,'Venta_hoy'] / df.loc[:,'Venta_uni_hoy']
df=df[['ar','Demanda_uni_equil']].groupby('ar',as_index=False).sum()
x=df.loc[:,'ar']
y=df.loc[:,'Demanda_uni_equil']
```

```
#plt.scatter(x, y,  color='black')
plt.plot(x, y,  color='blue')
#print(df)

elapsed_cpu = time.clock() - start_cpu
elapsed_real = time.time() - start_real
print('CPU idő:%s, Valós idő:%s'%(elapsed_cpu,elapsed_real) )
```

Kimenet:

```
CPU idő:28.342396, Valós idő:28.3402650356
```



Feladat:

Két véletlen vektor közti (lináris) kapcsolat erősségét a (Pearson) korrelációs együtthatóval lehet mérni. http://pandas.pydata.org/pandas-docs/stable/computation.htmlhttps://hu.wikipedia.org/wiki/Korrel%C3%A1ci%C3%B3 Számoljuk ki a train halmazon vagy egy részén, az Adjusted Demand oszlopnak a többi numerikus (nem Id jellegű) oszloppal vett korrelációs együtthatóját. Adjuk meg az eredményt táblázatban, a korrelációs együtthatók abszolútértéke szerint csökkenően.

Kód:

```
from __future__ import division
import pandas as pd
import numpy as np
import operator, re, time, random
start_cpu = time.clock()
start_real = time.time()

%matplotlib inline
import matplotlib.pyplot as plt
cols=['Semana','Ruta_SAK','Venta_uni_hoy','Venta_hoy','Dev_uni_proxima','Dev
_proxima','Demanda_uni_equil']
    #Nem ID oszlopok a trainben
df = pd.read_csv('./input/train.csv',usecols=cols)
#print(df.dtypes)
#print(df.corr())
ci=0
corrs_k=[]
corrs_v=[]
for c1 in cols:
    cj=0
```

```
    for c2 in cols:
        if(ci<cj):
            corr=df[c1].corr(df[c2])
            if(not np.isnan(corr)):
                corrs_k.append(c1+' - '+c2)
                corrs_v.append(corr)
        cj+=1
    ci+=1
corrs = dict(zip(corrs_k, corrs_v))
corrs = sorted(corrs.items(), key=lambda x: abs(x[1]), reverse=True)
for t in corrs:
    print(t)
elapsed_cpu = time.clock() - start_cpu
elapsed_real = time.time() - start_real
print('CPU idő:%s, Valós idő:%s'%(elapsed_cpu,elapsed_real) )
```

Kimenet:

```
('Venta_uni_hoy - Demanda_uni_equil', 0.99726527176417223)
('Venta_uni_hoy - Venta_hoy', 0.73367773002687386)
('Venta_hoy - Demanda_uni_equil', 0.73297336017483372)
('Dev_uni_proxima - Dev_proxima', 0.12942116632481751)
('Venta_uni_hoy - Dev_proxima', 0.061937898646103204)
('Venta_hoy - Dev_proxima', 0.057269413422541667)
('Ruta_SAK - Demanda_uni_equil', 0.051843816067766169)
('Ruta_SAK - Venta_uni_hoy', 0.051319105211192519)
('Dev_proxima - Demanda_uni_equil', 0.035126899967845154)
('Ruta_SAK - Venta_hoy', 0.022922354008533211)
('Venta_uni_hoy - Dev_uni_proxima', 0.0087692896337227093)
('Venta_hoy - Dev_uni_proxima', 0.0053830116285123091)
('Dev_uni_proxima - Demanda_uni_equil', 0.0041349435554030381)
('Semana - Venta_uni_hoy', 0.0032285900087220465)
('Semana - Demanda_uni_equil', 0.0030001193972107364)
('Semana - Ruta_SAK', -0.0029242125574354522)
('Ruta_SAK - Dev_proxima', -0.0018301282939609815)
('Semana - Dev_proxima', 0.0014315917018908313)
('Semana - Venta_hoy', 0.0012959757286495695)
('Semana - Dev_uni_proxima', 0.00045772042450208258)
('Ruta_SAK - Dev_uni_proxima', 0.00036890259380236427)
CPU idő:159.640776, Valós idő:65.5971791744
```

Feladat

Tegnap meghirdettek egy új versenyt, ami mobil adatok alapján jósolja meg a felhasználó korát és nemét. Ezzel fogunk majd foglalkozni. Nézzétek át a verseny kiírását és a táblák jellemzőit gyűjtsétek ki mennyi a rekordok száma, különböző értékek száma oszloponként, hiányzó adatok oszloponként, mennyi a fiúk-lányok aránya a teljes adatállományban, készítsetek táblázatot és grafikon, mennyi különböző személynek k az életkora, ahol k=0-100. https://www.kaggle.com/c/talkingdata-mobile-user-demographics/data

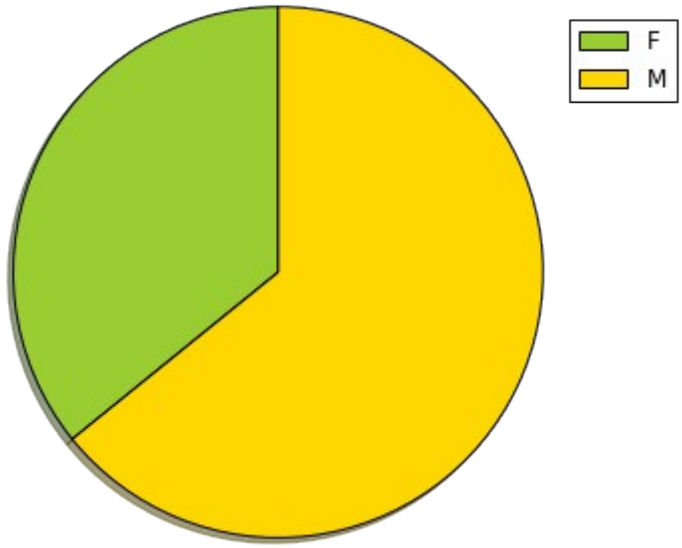
Kód:

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

from __future__ import division
import pandas as pd
import numpy as np
import operator, re, time, random
```

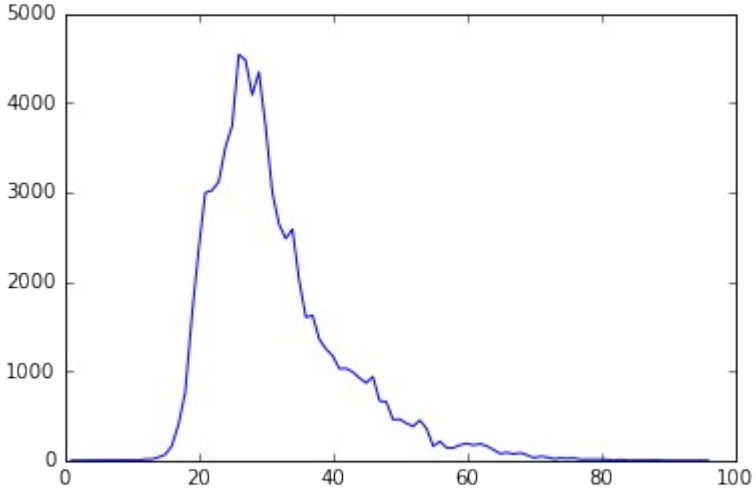
```
start_cpu = time.clock()
start_real = time.time()
df = pd.read_csv('./mobil/gender_age_train.csv')
#print(df)

print('Nem szerinti megoszlás:\n')
df_nem=df[['gender','age']].groupby(['gender'],as_index=False).count().rename(co
lumns={'age':'cnt'})
labels=df_nem.gender.values
sizes=df_nem.cnt.values
print(df_nem)
colors = ['yellowgreen', 'gold', 'lightskyblue', 'lightcoral']
patches, texts = plt.pie(sizes, colors=colors, shadow=True, startangle=90)
plt.legend(patches, labels, loc="best")
plt.axis('equal')
plt.tight_layout()
plt.show()

print('Életkor szerinti megoszlás:\n')
df_kor=df[['gender','age']].groupby(['age'],as_index=False).count().rename(colum
ns={'gender':'cnt'})
x=df_kor.loc[:,'age']
y=df_kor.loc[:,'cnt']
#print(x)
#print(y)
plt.plot(x, y,  color='blue')
plt.show()

print('Nem és életkor csoport szerinti megoszlás:\n')
df_csop=df[['gender','group']].groupby(['group'],as_index=False).count().rename(
columns={'gender':'cnt'})
labels=df_csop.group.values
sizes=df_csop.cnt.values
print(df_csop)
a=np.random.random(40)
from matplotlib import cm
cs=cm.Set1(np.arange(40)/40.)
patches, texts = plt.pie(sizes, colors=cs, shadow=True, startangle=90)
plt.legend(patches, labels, loc="best")
plt.axis('equal')
plt.tight_layout()
plt.show()

elapsed_cpu = time.clock() - start_cpu
elapsed_real = time.time() - start_real
print('CPU idő:%s, Valós idő:%s'%(elapsed_cpu,elapsed_real) )
```

Kimenet

```
Nem szerinti megoszlás:

  gender    cnt
0      F  26741
1      M  47904
```

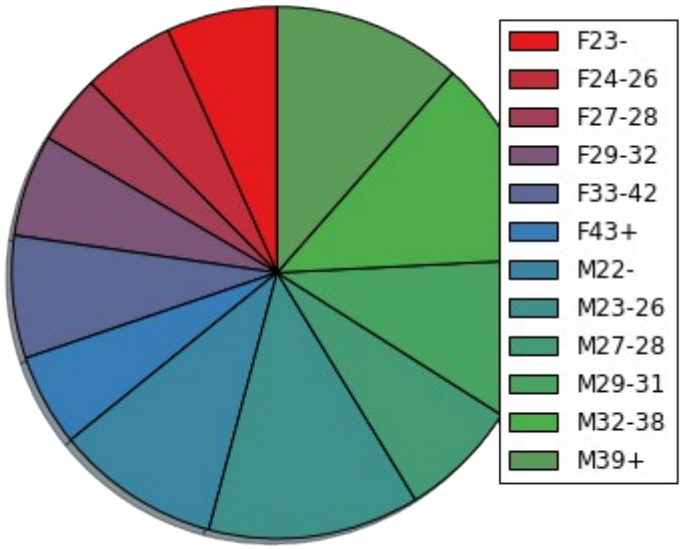


Életkor szerinti megoszlás:



Nem és életkor csoport szerinti megoszlás:

	group	cnt
0	F23-	5050
1	F24-26	4190
2	F27-28	3118
3	F29-32	4628
4	F33-42	5561
5	F43+	4194
6	M22-	7488
7	M23-26	9605
8	M27-28	5445
9	M29-31	7309
10	M32-38	9476
11	M39+	8581



CPU idő:0.604466, Valós idő:0.603837966919