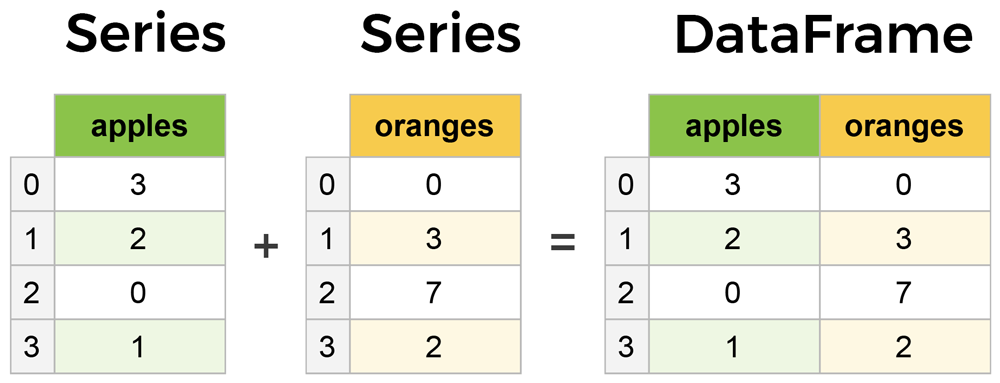
Python adatelemzés – SQL és Dataframe

# Pandas

Pandas alapok 10 percben: <http://pandas.pydata.org/pandas-docs/stable/10min.html>

Alap adatszerkezetek

* Series: 1 dimenziós indexelt tömb. A sorozat egyes elemei a címkéken (label) keresztül érhetők el.
* DataFrame: Excel táblához hasonló tábla, 2 dimenziós tömb. Sorok számozottak, oszlopokat nevekkel érjük el.



DataFrame létrehozás példa:

data = {

'apples': [3, 2, 0, 1],

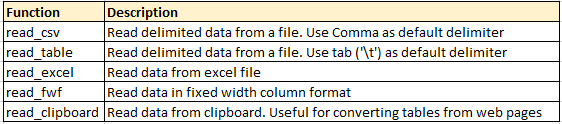
'oranges': [0, 3, 7, 2]

}

purchases = pd.DataFrame(data)

purchases

### Adatfájlok beolvasása



Excelből:

df=pd.read\_excel("excel.xlsx", "Munkalap1")

print df

Szövegfájl, tabulátor elválasztással oszlopnév megadással: df=pd.read\_csv("Test.csv",sep='\t', names=['UID', 'First Name', 'Last Name', 'Age'])

Sorok, oszlopok száma:

df.shape

Oszlopnevek lekérése: df.columns

Oszlop átnevezése:

df.rename(columns={

'Runtime (Minutes)': 'Runtime',

'Revenue (Millions)': 'Revenue\_millions'

}, inplace=True)

df.columns

Több oszlop átnevezése esetén:

df.columns = ['rank', 'genre', 'description', 'director', 'actors', 'year', 'runtime',

'rating', 'votes', 'revenue\_millions', 'metascore']

df.columns

## SQL 🡪 Pandas

A szükséges könyvtárak importálása és az adatset betöltése:

import numpy as np  
import pandas as pd  
titanic = pd.read\_csv("titanic.csv", dtype={"Age": np.float64} )

**Adatstruktúra:**

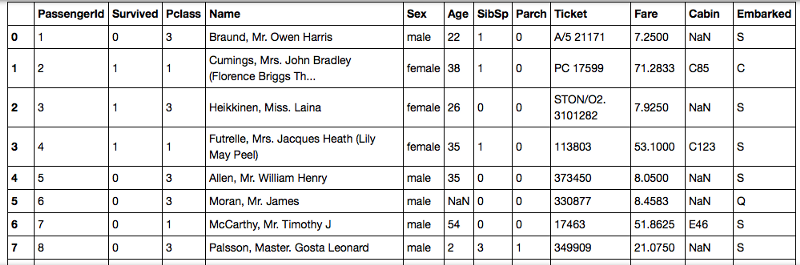
|  |  |
| --- | --- |
| **SQL** | **Pandas** |
| describe titanic -- Oracle | titanic.info() |

Eredménye:   
<class 'pandas.core.frame.DataFrame'>  
Int64Index: 891 entries, 0 to 890  
Data columns (total 12 columns):  
PassengerId 891 non-null int64  
Survived 891 non-null int64  
Pclass 891 non-null int64  
Name 891 non-null object  
Sex 891 non-null object  
Age 714 non-null float64  
SibSp 891 non-null int64  
Parch 891 non-null int64  
Ticket 891 non-null object  
Fare 891 non-null float64  
Cabin 204 non-null object  
Embarked 889 non-null object  
dtypes: float64(2), int64(5), object(5)  
memory usage: 90.5+ KB

#### SQL – SELECT

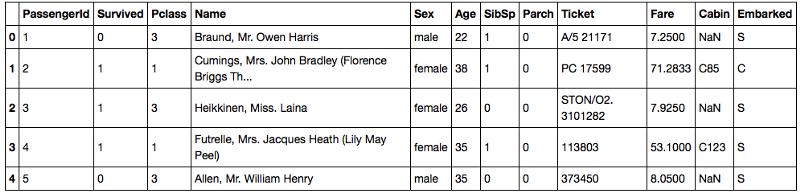
**SELECT \* FROM**

|  |  |
| --- | --- |
| **SQL** | **Pandas** |
| select \* from titanic; | titanic |



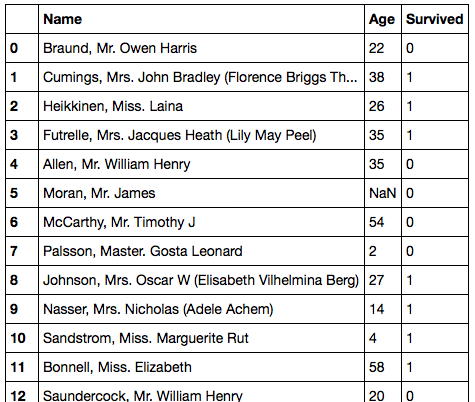
**Csak néhány sor kiválasztása**

|  |  |
| --- | --- |
| **SQL** | **Pandas** |
| select \* from titanic where WHERE rownum <= 5; -- Oracle  select top 5 \* from titanic; -- SQL server | titanic.head(5) - első 5 sor  titanic.head - első sorok (5)  titanic.tail - utolsó sorok (5)  Vagy indexeléssel (sor és oszlop indexek megadásával:  titanic.iloc[0:5,:]  titanic.iloc[:5,:] |



**Oszlopok kiválasztása**

|  |  |
| --- | --- |
| **SQL** | **Pandas** |
| select Name,Age,Survived from titanic; | titanic[["Name","Age","Survived"]]  titanic.loc[:,["Name","Age","Survived"]] |

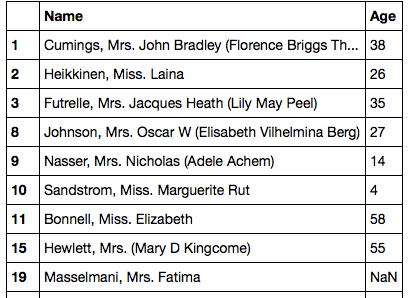


**SELECT WHERE**

|  |  |
| --- | --- |
| **SQL** | **Pandas** |
| select \* from titanic where Survived = 1; | titanic[titanic["Survived"]==1] |

**Összetett SELECT – Adjuk meg a nemét és az életkorát a hölgy túlélőknek**

|  |  |
| --- | --- |
| **SQL** | **Pandas** |
| select Name,Age  from titanic  where Survived = 1 and Sex='female'; | titanic[(titanic["Survived"]==1) & (titanic["Sex"]=="female")][["Name","Age"]] |



**Aggregátor függvények - számoljuk meg, hogy hány túlélő volt és hányan nem élték túl**

|  |  |
| --- | --- |
| **SQL** | **Pandas** |
| select survived, count(\*)  from titanic  group by survived; | titanic.groupby(“Survived”).size() |

További példák:

<https://pandas.pydata.org/pandas-docs/stable/comparison_with_sql.html>

## Pandas – egyebek

Adatok konvertálása

srting\_outcome = str(numeric\_input)

integer\_outcome = int(string\_input)

float\_outcome = float(string\_input)

Dátum konvertálás

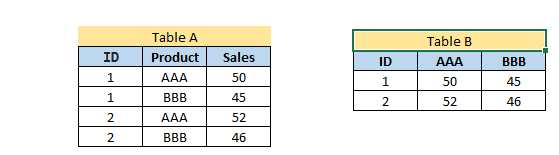
from datetime import datetime

char\_date = 'Apr 1 2015 1:20 PM'

date\_obj = datetime.strptime(char\_date, '%b %d %Y %I:%M%p')

print date\_obj

Táblra transzponálása (egy oszlop szerint) hasonlóan az Excel Pivot táblájához:



df=pd.read\_excel("tabla.xlsx", "Munkalap1")

print df

result= df.pivot(index= 'ID', columns='Product', values='Sales')

result

Rendezés:

print df.sort(['Product','Sales'], ascending=[True, False])

Random 5 adat kiválasztása:

import numpy as np

import pandas as pd

from random import sample

rindex = np.array(sample(xrange(len(df)), 5)) #random index

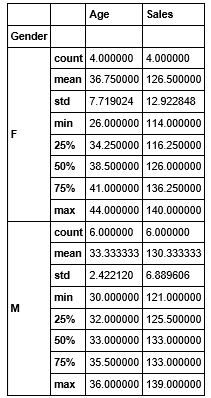
dfr = df.ix[rindex] #ennek az 5 elemnek, mint sorszámnak a kiválasztása

print dfr

Group By

test= df.groupby(['Gender'])

test.describe()



Hiányzó értékek:

df.isnull()