Zeppelin Notebook anonymous

assign4_polution

default

FINISHED

```
val polutionData = sqlContext.read.format("com.databricks.spark.csv").option("header", "true").load("/home/scarface/desktop/sem-3/c
```

```
polutionData: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [ozone: string, particullate_matter: string ... 6 more fields
]
```

Took 45 sec. Last updated by anonymous at February 15 2017, 1:53:48 AM.

FINISHED

```
polutionData.toDF().registerTempTable("polData")
```

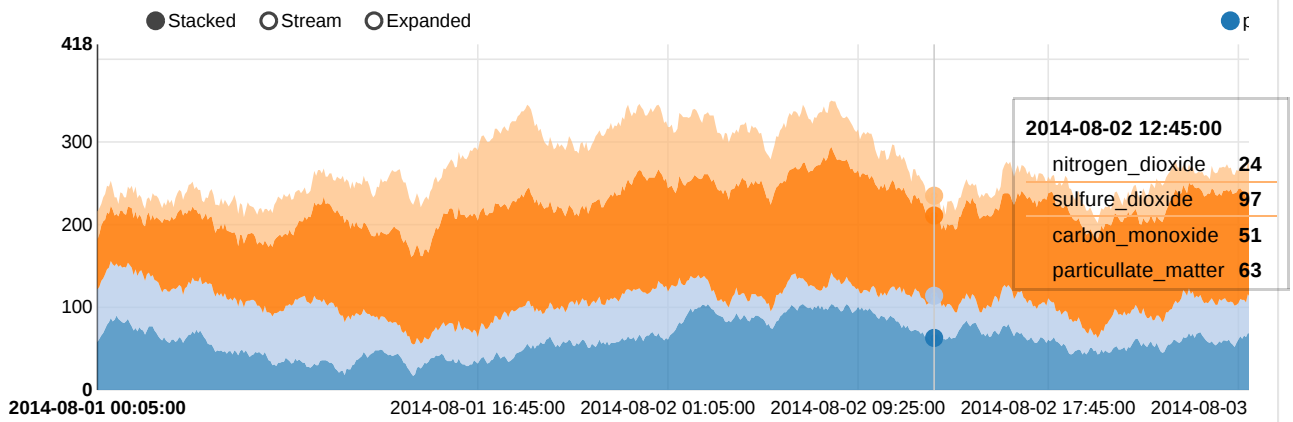
```
warning: there was one deprecation warning; re-run with -deprecation for details
```

Took 1 sec. Last updated by anonymous at February 15 2017, 1:53:53 AM.

FINISHED

```
%sql
select * from polData
```

settings




```
Results are limited by 1000.
```

Took 11 sec. Last updated by anonymous at February 15 2017, 1:54:08 AM. (outdated)

FINISHED

```
%sql
select * from polData where timestamp like "2014-08-01%"
```

settings

All fields:

ozone | particullate_matter | carbon_monoxide | sulfure_dioxide | nitrogen_dioxide | longitude | latitude | timestamp

Keys

timestamp ✖

Zeppelin Notebook anonymous

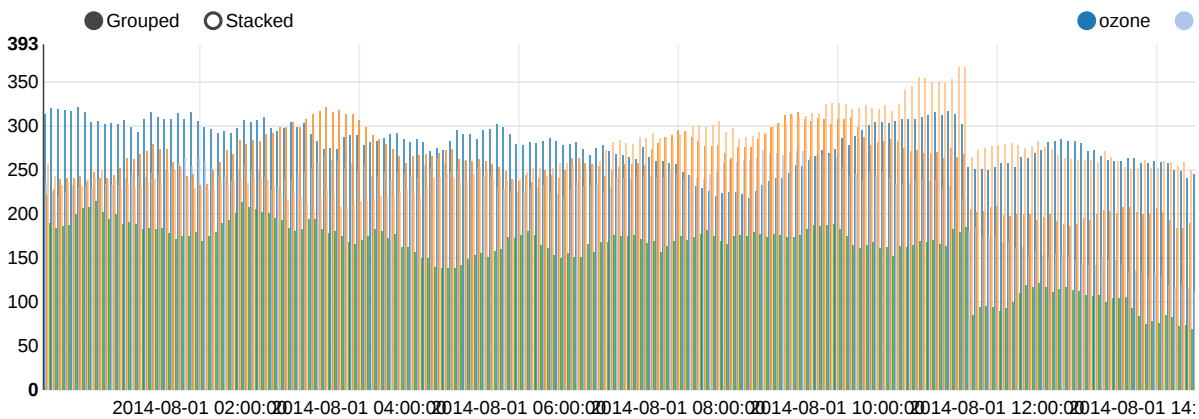
Groups

assign4_polution

default

Values

ozone SUM ✖
carbon_monoxide SUM ✖
sulfure_dioxide SUM ✖
nitrogen_dioxide SUM ✖
particullate_matter SUM ✖




```
Results are limited by 1000.
```

Took 1 sec. Last updated by anonymous at February 15 2017, 1:55:06 AM. (outdated)

ERROR

```
%sql
select * from polData
```

```
Table or view not found: polData; line 1 pos 14
set zeppelin.spark.sql.stacktrace = true to see full stacktrace
```

Took 42 sec. Last updated by anonymous at April 15 2017, 7:44:09 PM.

READY

FINISHED

```
%pyspark
import glob
files = glob.glob("/home/scarface/Desktop/sem-3/capstone/Prithvi/pollution/*.csv")
print (files[:20])
```

```
['/home/scarface/Desktop/sem-3/capstone/Prithvi/pollution/pollutionData192866.csv', '/home/scarface/Desktop/sem-3/capstone/Prithvi/p
ollution/pollutionData206475.csv', '/home/scarface/Desktop/sem-3/capstone/Prithvi/pollution/pollutionData193402.csv', '/home/scarfac
```

Zeppelin Notebook anonymous default

assign4_pollution

```
e/Desktop/sem-3/capstone/Prithvi/pollution/pollutionData195312.csv', '/home/scarface/Desktop/sem-3/capstone/Prithvi/pollution/pollut
ca201722.csv', '/home/scarface/Desktop/sem-3/capstone/Prithvi/pollution/pollutionData197896.csv', '/home/scarface/Desktop/sem-3
/capstone/Prithvi/pollution/pollutionData181331.csv', '/home/scarface/Desktop/sem-3/capstone/Prithvi/pollution/pollutionData201722.c
sv', '/home/scarface/Desktop/sem-3/capstone/Prithvi/pollution/pollutionData194960.csv', '/home/scarface/Desktop/sem-3/capstone/Prith
vi/pollution/pollutionData190709.csv', '/home/scarface/Desktop/sem-3/capstone/Prithvi/pollution/pollutionData190709.csv', '/home/sca
rface/Desktop/sem-3/capstone/Prithvi/pollution/pollutionData179228.csv', '/home/scarface/Desktop/sem-3/capstone/Prithvi/pollution/po
llutionData187006.csv', '/home/scarface/Desktop/sem-3/capstone/Prithvi/pollution/pollutionData187297.csv', '/home/scarface/Desktop/s
em-3/capstone/Prithvi/pollution/pollutionData185396.csv', '/home/scarface/Desktop/sem-3/capstone/Prithvi/pollution/pollutionData1929
46.csv', '/home/scarface/Desktop/sem-3/capstone/Prithvi/pollution/pollutionData158983.csv', '/home/scarface/Desktop/sem-3/capstone/P
rithvi/pollution/pollutionData197734.csv', '/home/scarface/Desktop/sem-3/capstone/Prithvi/pollution/pollutionData188039.csv', '/home
/scarface/Desktop/sem-3/capstone/Prithvi/pollution/pollutionData195552.csv']
```

Took 4 sec. Last updated by anonymous at April 15 2017, 7:46:52 PM.

FINISHED

```
%pyspark
import csv
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv(files[1])
corr = df[['ozone', 'particullate_matter', 'carbon_monoxide', 'sulfure_dioxide', 'nitrogen_dioxide']].corr()
#print (corr)
plt.matshow(corr)
plt.show()
```

Took 0 sec. Last updated by anonymous at April 15 2017, 8:34:12 PM.

FINISHED

```
%pyspark
import csv
dic={}
for each in files:
    oz=0
    pm=0
    cm=0
    sd=0
    nd=0
    with open(each, "rb") as f:
        reader = csv.reader(f)
        a=reader.next()
        i=0
        for r in reader:
            oz=oz+int(r[0])
            pm=pm+int(r[1])
            cm=cm+int(r[2])
            sd=sd+int(r[3])
            nd=nd+int(r[4])
            i=i+1
        if (r[5]=="10.10711200000003"):
            print i
            print oz
        ozA=oz/i
        pmA=pm/i
        cmA=cm/i
        sdA=sd/i
        ndA=nd/i

        dic[str(r[5])+"+"+str(r[6])]=[ozA,pmA,cmA,sdA,ndA]
print dic
#31
```

```
17568
1674447
17568
1856742
```

Took 26 sec. Last updated by anonymous at March 27 2017, 8:21:32 AM. (outdated)

FINISHED

```
%pyspark
```

Zeppelin Notebook anonymous

assign4_polution

default

```
['10.250786139881143+56.20251117218925', '10.149547918402732+56.170886798880105', '10.171611868717264+56.174917180024806', '10.10711200000003+56.21731711429131', '10.269519082174156+56.2107989833621']
```

Took 0 sec. Last updated by anonymous at March 27 2017, 9:20:15 AM. (outdated)

FINISHED

```
%pyspark
headLi=["ozone", "particullate_matter", "carbon_monoxide", "sulfure_dioxide", "nitrogen_dioxide"]
for l in li:
    latLon=l.split("+")
    print "Highest percentage of ozone levels are detected at location " + latLon[0] + ", "+ latLon[1]
```

```
Highest percentage of ozone levels are detected at location 10.250786139881143, 56.20251117218925
Highest percentage of ozone levels are detected at location 10.149547918402732, 56.170886798880105
Highest percentage of ozone levels are detected at location 10.171611868717264, 56.174917180024806
Highest percentage of ozone levels are detected at location 10.10711200000003, 56.21731711429131
Highest percentage of ozone levels are detected at location 10.269519082174156, 56.2107989833621
```

Took 0 sec. Last updated by anonymous at March 27 2017, 9:24:49 AM.

FINISHED

```
%pyspark
oxMi=100
pmMi=100
cmMi=100
sdMi=100
ndMi=100
o,p,c,s,n="","","","","",""
for j in dic:
    if dic[j][0] < oxMi:
        o=j
        oxMi=dic[j][0]
    if dic[j][1] < pmMi:
        p=j
        pmMi=dic[j][1]
    if dic[j][2] < cmMi:
        c=j
        cmMi=dic[j][2]
    if dic[j][3] < sdMi:
        s=j
        sdMi=dic[j][3]
    if dic[j][4] < ndMi:
        n=j
        ndMi=dic[j][4]
lim= [o,p,c,s,n]
print lim
```

```
['10.173023480985648+56.21071820426365', '10.141663381614649+56.124304277059785', '10.175338207340246+56.179322409643085', '10.149874309192683+56.148242094184255', '10.212459373016372+56.18696229571611']
```

Took 1 sec. Last updated by anonymous at March 27 2017, 9:42:40 AM.

Zeppelin Notebook anonymous

assign4_polution

FINISHED default

```
%pyspark
headLi=["ozone", "particullate_matter", "carbon_monoxide", "sulfure_dioxide", "nitrogen_dioxide"]
for l in lim:
    latLon=l.split("+")
    print "Lowest percentage of ozone levels are detected at location " + latLon[0] + ", "+ latLon[1]
#87
```

```
Lowest percentage of ozone levels are detected at location 10.173023480985648, 56.21071820426365
Lowest percentage of ozone levels are detected at location 10.141663381614649, 56.124304277059785
Lowest percentage of ozone levels are detected at location 10.175338207340246, 56.179322409643085
Lowest percentage of ozone levels are detected at location 10.149874309192683, 56.148242094184255
Lowest percentage of ozone levels are detected at location 10.212459373016372, 56.18696229571611
```

Took 0 sec. Last updated by anonymous at March 27 2017, 9:45:06 AM. (outdated)

FINISHED

```
%pyspark
print headLi
print li
print lim
```

```
['ozone', 'particullate_matter', 'carbon_monoxide', 'sulfure_dioxide', 'nitrogen_dioxide']
['10.250786139881143+56.20251117218925', '10.149547918402732+56.170886798880105', '10.171611868717264+56.174917180024806', '10.10711200000003+56.21731711429131', '10.269519082174156+56.2107989833621']
['10.173023480985648+56.21071820426365', '10.141663381614649+56.124304277059785', '10.175338207340246+56.179322409643085', '10.149874309192683+56.148242094184255', '10.212459373016372+56.18696229571611']
```

Took 0 sec. Last updated by anonymous at March 27 2017, 9:49:08 AM.

FINISHED

```
%pyspark
from sklearn import datasets, linear_model
```

Took 5 sec. Last updated by anonymous at March 27 2017, 10:05:04 AM.

FINISHED

```
%pyspark
ozLi=[]
pmLi=[]
cmLi=[]
sdLi=[]
ndLi=[]
with open(files[0], "rb") as f:
    reader = csv.reader(f)
    a=reader.next()
    for row in reader:
        ozLi.append(int(row[0]))
        pmLi.append(int(row[1]))
        cmLi.append(int(row[2]))
        sdLi.append(int(row[3]))
        ndLi.append(int(row[4]))
print ozLi[:100]
```

```
[30, 29, 34, 35, 34, 39, 43, 41, 41, 39, 35, 37, 35, 38, 37, 38, 42, 37, 38, 38, 37, 33, 36, 36, 37, 35, 40, 38, 36, 31, 31, 31, 31, 26, 27, 26, 26, 31, 35, 31, 34, 39, 40, 40, 41, 36, 37, 37, 37, 33, 31, 36, 32, 33, 33, 38, 39, 37, 37, 32, 34, 39, 42, 46, 44, 39, 35, 31, 26, 25, 21, 25, 23, 19, 28, 32, 32, 31, 31, 26, 27, 26, 26, 25, 28, 30, 35, 37, 41, 40, 43, 47, 44, 42, 37, 35, 39, 42, 47, 49]
```

Took 0 sec. Last updated by anonymous at March 27 2017, 10:43:10 AM.

FINISHED

```
%pyspark
sd_train=ozLi[:-20]
sd_test=ozLi[-20:]

pm_train=ozLi[:-20]
pm_test=ozLi[-20:]

pm_train=pd.DataFrame(pm_train)
```

Zeppelin Notebook anonymous

Took 0 sec. Last updated by anonymous at March 27 2017, 10:35:11 AM.

assign4_polution

default

FINISHED

```
%pyspark
reg = linear_model.LinearRegression()
reg.fit(sd_train, pm_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

Took 0 sec. Last updated by anonymous at March 27 2017, 10:35:15 AM.

FINISHED

```
%pyspark
import numpy as np
print('Coefficients: ', reg.coef_)
print("Mean squared error: 35.07" % np.mean((reg.predict(sd_train) - pm_train) ** 2))
```

```
('Coefficients: ', array([[ 1.]]))
Mean squared error: 35.07
```

Took 0 sec. Last updated by anonymous at March 27 2017, 10:43:38 AM.

READY