Zeppelin Notebook

classwork/data_grp

Search over Notebooks

anonymous

Untitled

default

FINISHED

```
%pyspark
import timeit
col = ['Identification',  'Salesprice' , 'Finishedsquarefeet', 'Numberofbedrooms' ,'Number
'Quality' , 'style', 'Lotsize' , 'Adjacenttohighway']
```

FINISHED

```
%pyspark
start = timeit.timeit()
data1 = pd.read_csv('/Users/neha/Desktop/real.csv')
data1.columns  = col
end = timeit.timeit()
print(end - start)
```

```
0.0016028881073
```

FINISHED

```
%pyspark
import timeit
start = timeit.timeit()
print("hello")
end = timeit.timeit()
print(end - start)
```

```
hello
-0.00362491607666
```

FINISHED

```
%pyspark
data1.head()
```

```
   Identification  Salesprice  Finishedsquarefeet  Numberofbedrooms  \
0               2      340000                2058                 4
1               3      250000                1780                 4
2               4      205500                1638                 4
3               5      275500                2196                 4
4               6      248000                1966                 4
   Numberofbathrooms  Airconditioning  Garagesize  Pool  Yearbuilt  Quality  \
0                  2                1           2     0       1976        2
1                  3                1           2     0       1980        2
2                  2                1           2     0       1963        2
3                  3                1           2     0       1968        2
4                  3                1           5     1       1972        2
   style  Lotsize  Adjacenttohighway
0      1    22912                  0
1      1    21345                  0
2      1    17342                  0
3      7    21786                  0
4      1    18902                  0
```

```
%pyspark
from pandas import Series, DataFrame
import pandas as pd
import numpy as np
```

FINISHED

```
%pyspark
start = timeit.timeit()
grouped = data1.groupby('Yearbuilt')
end = timeit.timeit()
print(end - start)
```

FINISHED

```
-0.00120687484741
```

```
%pyspark
start = timeit.timeit()
data1.info()
end = timeit.timeit()
print(end - start)
```

FINISHED

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 521 entries, 0 to 520
Data columns (total 13 columns):
Identification        521 non-null int64
Salesprice            521 non-null int64
Finishedsquarefeet    521 non-null int64
Numberofbedrooms      521 non-null int64
Numberofbathrooms     521 non-null int64
Airconditioning       521 non-null int64
Garagesize            521 non-null int64
Pool                  521 non-null int64
Yearbuilt             521 non-null int64
Quality               521 non-null int64
style                 521 non-null int64
Lotsize               521 non-null int64
Adjacenttohighway     521 non-null int64
dtypes: int64(13)
memory usage: 53.0 KB
```

FINISHED

```
%pyspark
start = timeit.timeit()
data1[-4:]
end = timeit.timeit()
print(end - start)
```

```
-0.00600600242615
```

FINISHED

```
%pyspark
start = timeit.timeit()
lotsize_corr = lambda x: x.corrwith(x['Lotsize'])
end = timeit.timeit()
print(end - start)
```

```
-0.00641107559204
```

FINISHED

```
%pyspark
start = timeit.timeit()
import statsmodels.api as sm
def regression(data, yvar, xvars):
  Y = data[yvar]
  X = data[xvars]
  X['intercept'] = 1.
  result = sm.OLS(Y,X).fit()
  return result.params
  end = timeit.timeit()
print(end - start)
```

```
-0.0117671489716
```

```
%pyspark
start = timeit.timeit()
xvar = [ 'Finishedsquarefeet', 'Numberofbedrooms' ,'Numberofbathrooms', 'Airconditioning',
'Adjacenttohighway']
end = timeit.timeit()
print(end - start)
```

FINISHED

```
-0.00342702865601
```

```
%pyspark
start = timeit.timeit()
by_Numberofbathrooms = data1.groupby('Numberofbathrooms')
end = timeit.timeit()
print(end - start)
```

FINISHED

```
-0.00147581100464
```

```
%pyspark
start = timeit.timeit()

by_Numberofbathrooms.apply(regression,'Salesprice',xvar)
end = timeit.timeit()
print(end - start)
```

FINISHED

```
0.00366377830505
```

```
%pyspark

by_Numberofbathrooms.apply(regression,'Salesprice',xvar)
```

FINISHED

	Finishedsquarefeet	Numberofbedrooms	Numberofbathrooms	Numberofbathrooms
0	0.799334	0.000000	0.000000e+00	
1	41.704401	7789.667268	-3.624188e+05	
2	89.617662	7156.517226	-5.236697e+05	
3	117.966949	-8478.624895	-5.865658e+05	
4	112.261047	469.580325	-1.217455e+06	
5	217.747483	-24555.066427	-4.799156e+06	
6	0.852219	0.001028	1.028216e-03	
7	79.841580	0.093354	2.869320e-01	

	Airconditioning	Garagesize	Pool	Yearbuilt	Numberofbathrooms
0	0.000375	0.001126	0.000000	0.747897	
1	3884.646014	5832.481388	9457.310636	422.877244	
2	11422.173069	-5425.120839	14027.973629	666.404923	
3	-25415.981403	2418.531835	28820.314519	1137.685472	
4	-81398.314280	1412.029447	39438.656304	2758.055840	
5	-57923.740050	1084.397776	41449.484623	12554.121394	

FINISHED

READY