Analysis of LEGO brick prices over the years with PySpark

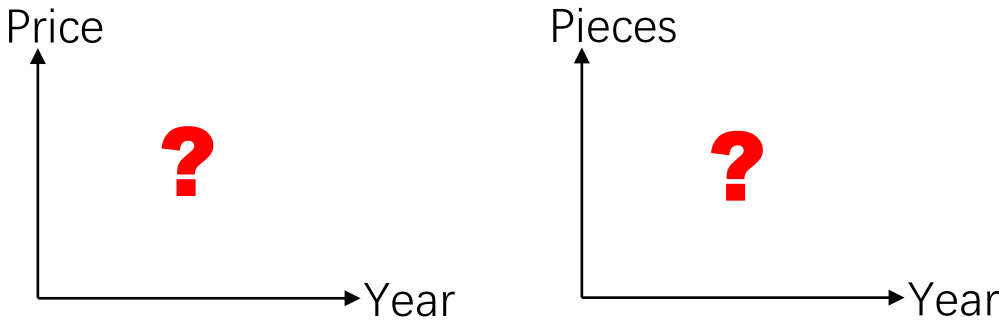
BigData Projekt

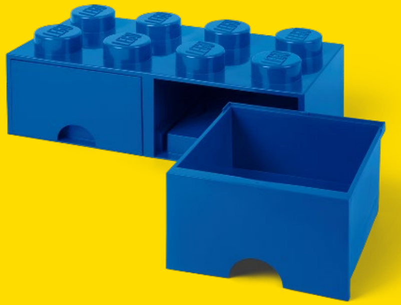
Guo Chengcheng

Aim

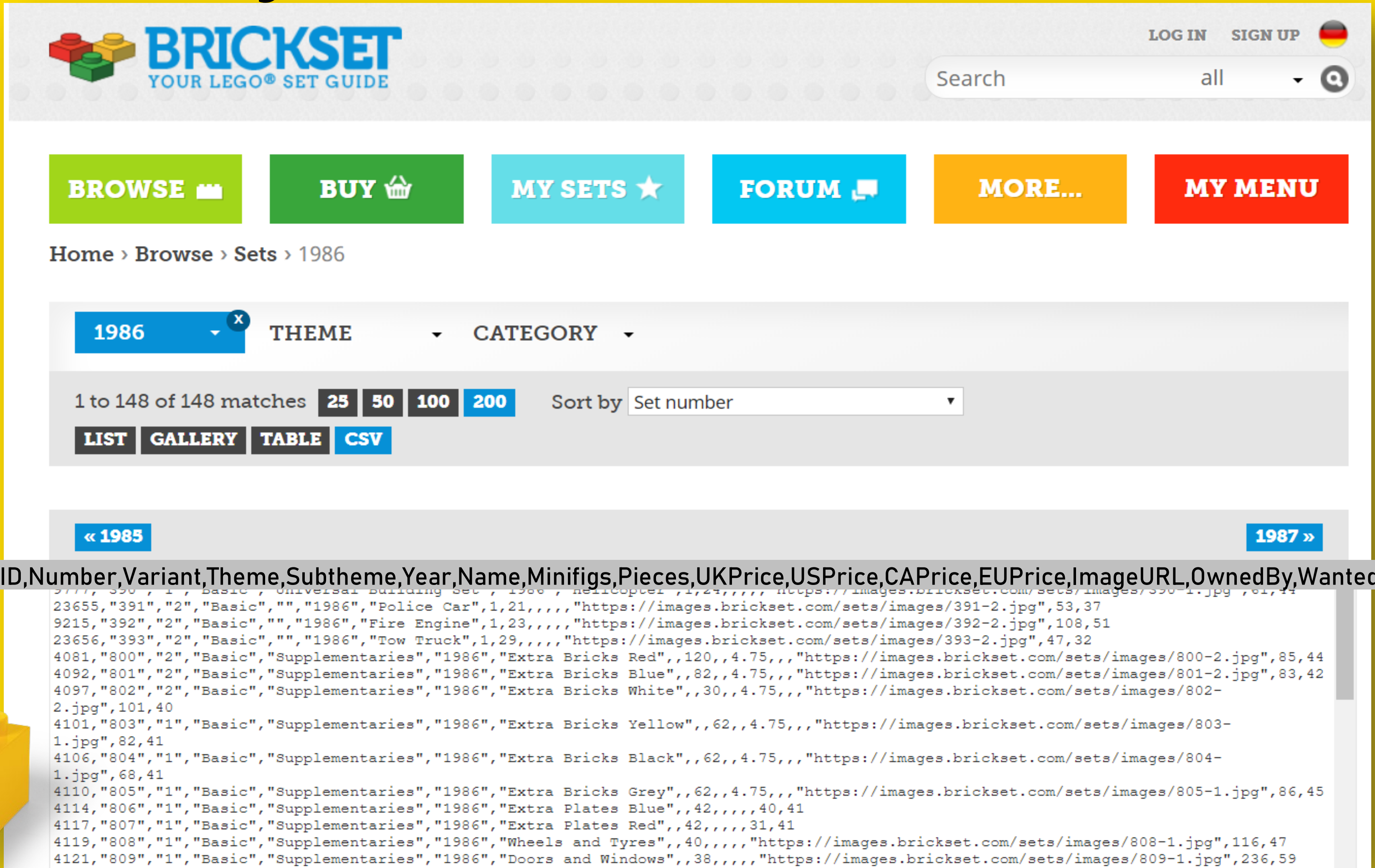
to evaluate the price history of LEGO sets

and the Mass of pasts LEGO sets



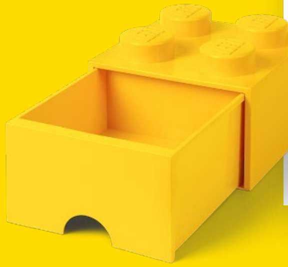


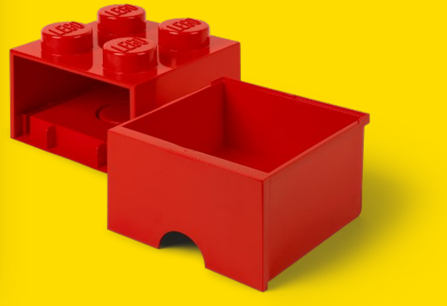
How is my dataset



SetID,Number,Variant,Theme,Subtheme,Year,Name,Minifigs,Pieces,UKPrice,USPrice,CAPrice,EUPrice,ImageURL,OwnedBy,WantedBy

```
23655,"391","2","Basic","","1986","Police Car",1,21,,,,,"https://images.brickset.com/sets/images/391-2.jpg",53,37
9215,"392","2","Basic","","1986","Fire Engine",1,23,,,,,"https://images.brickset.com/sets/images/392-2.jpg",108,51
23656,"393","2","Basic","","1986","Tow Truck",1,29,,,,,"https://images.brickset.com/sets/images/393-2.jpg",47,32
4081,"800","2","Basic","Supplementaries","1986","Extra Bricks Red",,120,,4.75,,,"https://images.brickset.com/sets/images/800-2.jpg",85,44
4092,"801","2","Basic","Supplementaries","1986","Extra Bricks Blue",,82,,4.75,,,"https://images.brickset.com/sets/images/801-2.jpg",83,42
4097,"802","2","Basic","Supplementaries","1986","Extra Bricks White",,30,,4.75,,,"https://images.brickset.com/sets/images/802-
2.jpg",101,40
4101,"803","1","Basic","Supplementaries","1986","Extra Bricks Yellow",,62,,4.75,,,"https://images.brickset.com/sets/images/803-
1.jpg",82,41
4106,"804","1","Basic","Supplementaries","1986","Extra Bricks Black",,62,,4.75,,,"https://images.brickset.com/sets/images/804-
1.jpg",68,41
4110,"805","1","Basic","Supplementaries","1986","Extra Bricks Grey",,62,,4.75,,,"https://images.brickset.com/sets/images/805-1.jpg",86,45
4114,"806","1","Basic","Supplementaries","1986","Extra Plates Blue",,42,,,,,40,41
4117,"807","1","Basic","Supplementaries","1986","Extra Plates Red",,42,,,,,31,41
4119,"808","1","Basic","Supplementaries","1986","Wheels and Tyres",,40,,,,,"https://images.brickset.com/sets/images/808-1.jpg",116,47
4121,"809","1","Basic","Supplementaries","1986","Doors and Windows",,38,,,,,"https://images.brickset.com/sets/images/809-1.jpg",236,59
```

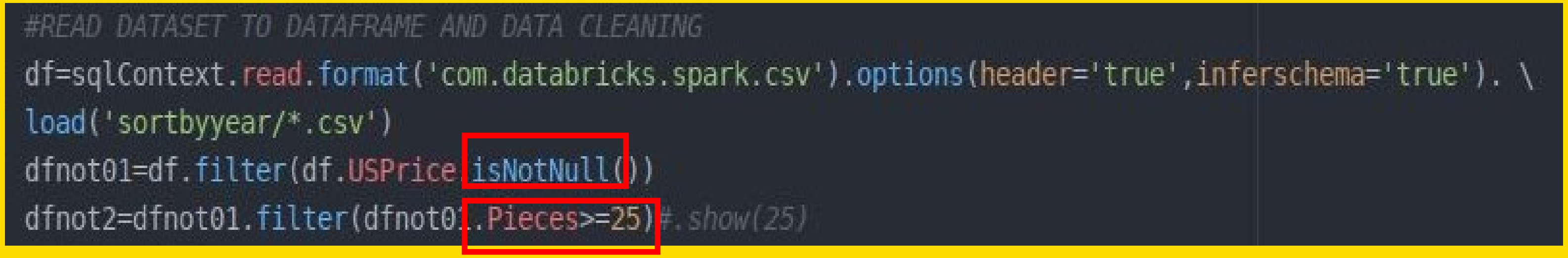




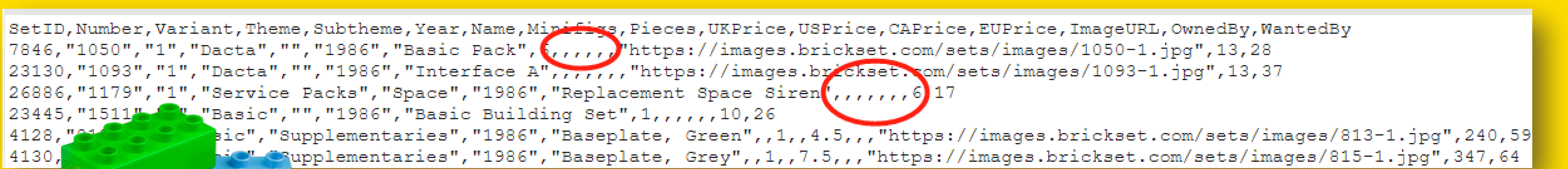
LEGO

1st Step

DataCleaning

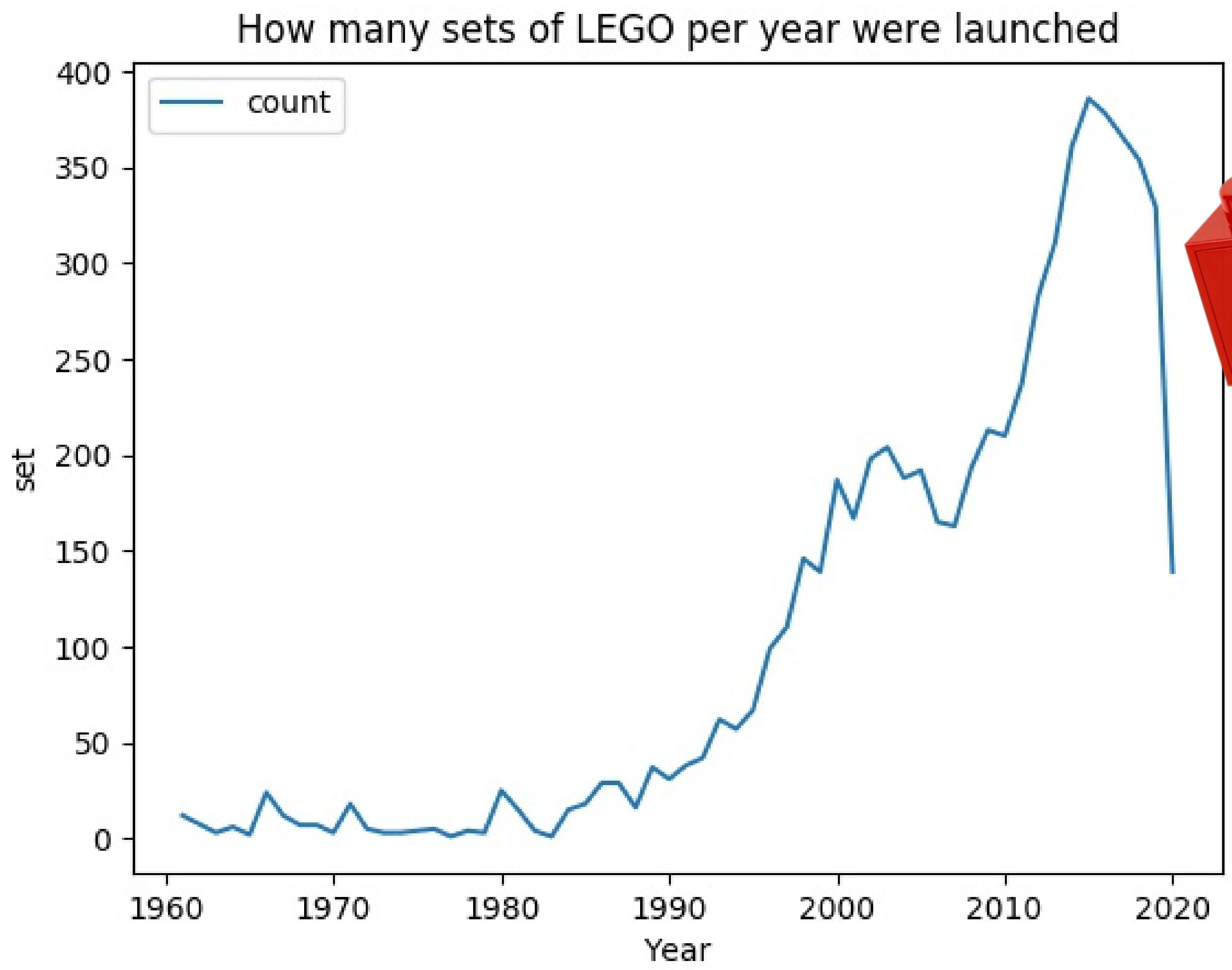


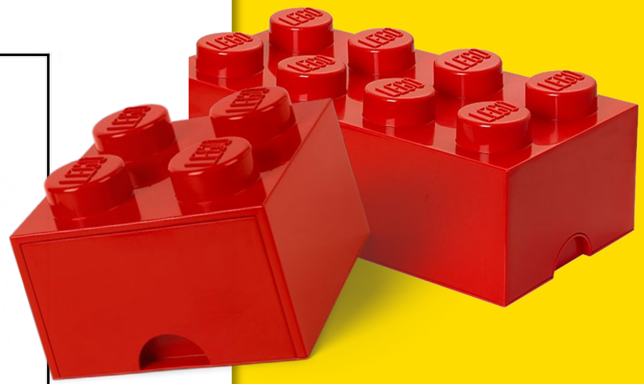
```
#READ DATASET TO DATAFRAME AND DATA CLEANING
df=sqlContext.read.format('com.databricks.spark.csv').options(header='true',inferschema='true'). \
load('sortbyyear/*.csv')
dfnot01=df.filter(df.USPrice.isNotNull())
dfnot2=dfnot01.filter(dfnot01.Pieces>=25)#.show(25)
```



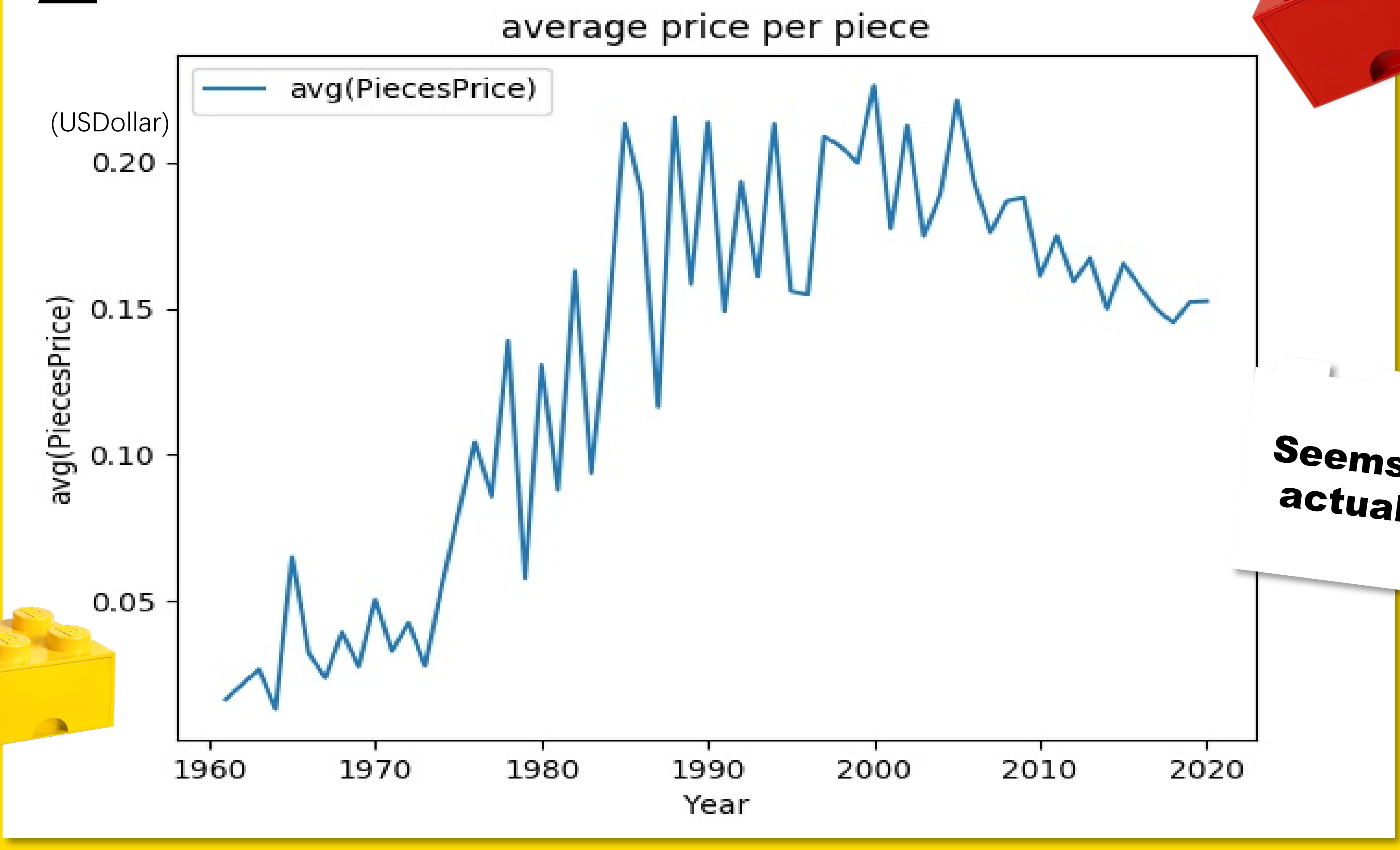
```
SetID,Number,Variant,Theme,Subtheme,Year,Name,Minifigs,Pieces,UKPrice,USPrice,CAPrice,EUPrice,ImageURL,OwnedBy,WantedBy
7846,"1050","1","Dacta","","1986","Basic Pack",,,,,,,"https://images.brickset.com/sets/images/1050-1.jpg",13,28
23130,"1093","1","Dacta","","1986","Interface A",,,,,,,"https://images.brickset.com/sets/images/1093-1.jpg",13,37
26886,"1179","1","Service Packs","Space","1986","Replacement Space Siren",,,,,,,,6,17
23445,"1511[illegible]"Basic","","1986","Basic Building Set",1,,,,,,10,26
4128,"[illegible]sic","Supplementaries","1986","Baseplate, Green",,1,,4.5,,,"https://images.brickset.com/sets/images/813-1.jpg",240,59
4130,[illegible]"Supplementaries","1986","Baseplate, Grey",,1,,7.5,,,"https://images.brickset.com/sets/images/815-1.jpg",347,64
```

Plot 1


How many sets of LEGO per year were launched
count
400
350
300
250
200
150
100
50
0
set
1960
1970
1980
1990
2000
2010
2020
Year

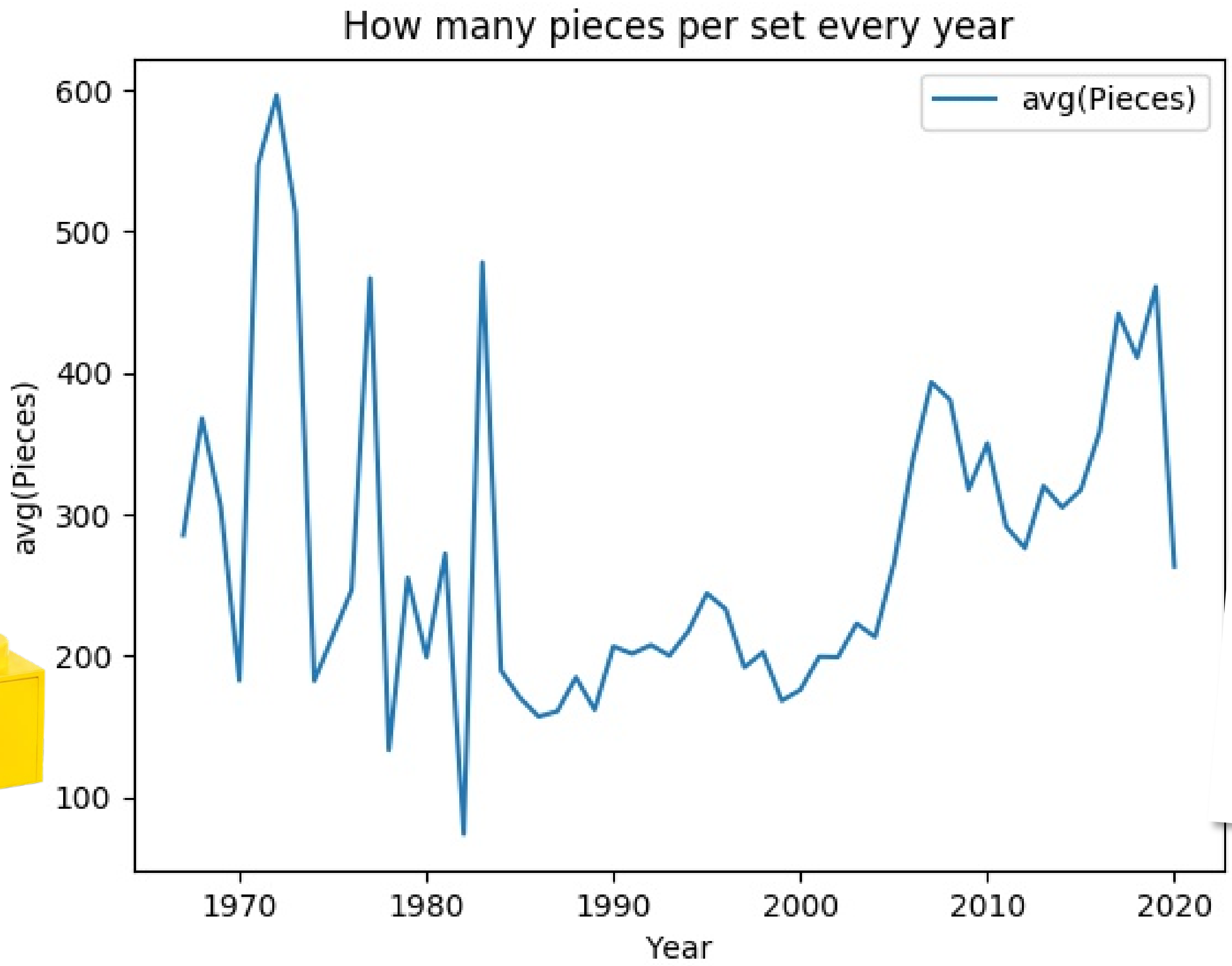


Plot 2



Seems to be increased , actually not

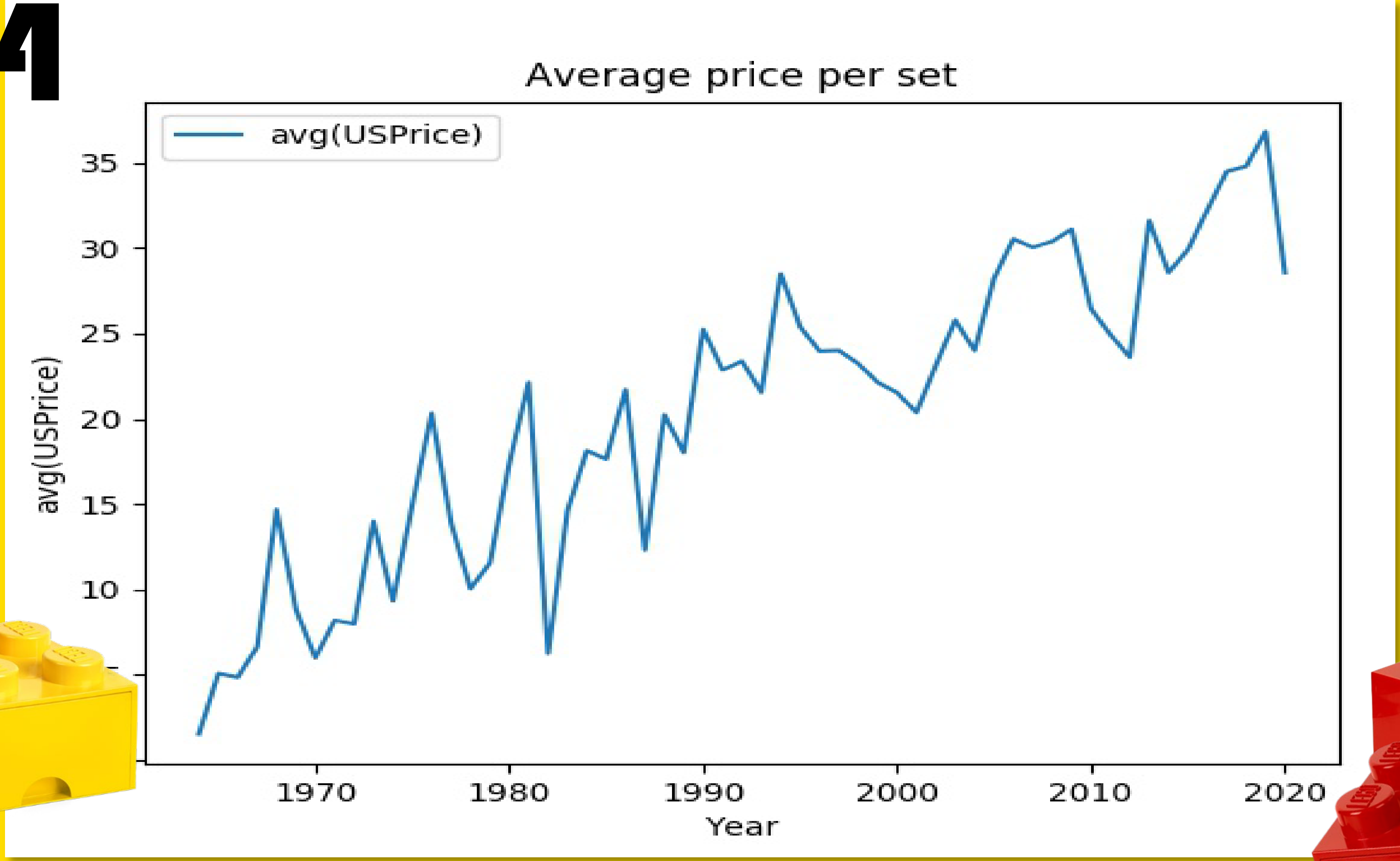
Plot 3



2007:Star Wars-Ultimate Collector's Millennium Falcon 5197

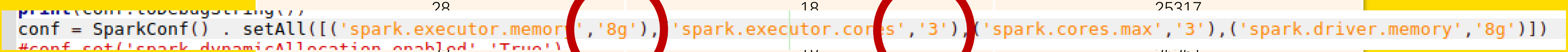
2008:Taj Mahal 5922

Plot 4

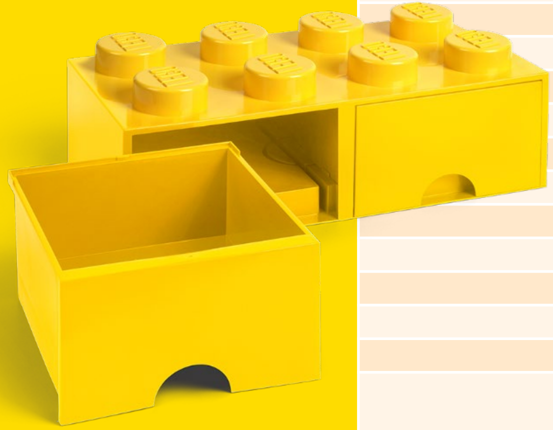

Average price per set
avg(USPrice)
35
30
25
20
15
10
avg(USPrice)
1970
1980
1990
2000
2010
2020
Year

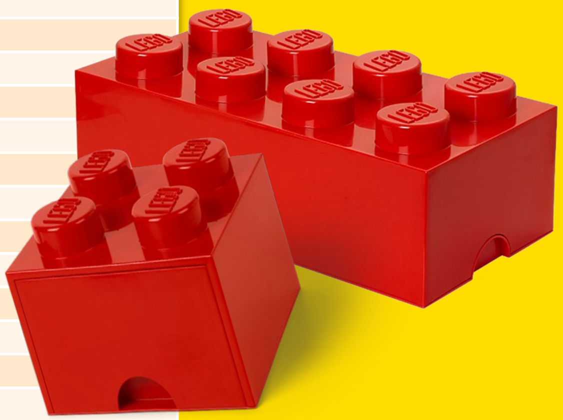
5.Benchmark Results

memory	executor cores	time
8	3	32096
80	9	26789
64	9	26416
30	9	27499
28	9	25703
40	9	27741
28	10	28492
16	9	27284
20	16	26847
45	16	26011
80	16	28068
64	18	27333
30	18	24802
28	18	25317
[illegible]	[illegible]	[illegible]
22	18	25361
8	20	26799
100	30	25031
3	30	25758
50	30	26098
1	30	26708
8	30	27467
5	30	29180
102	32	27028
100	32	27525
30	36	24346
24	36	24734
10	36	24215
20	36	24734
1	36	25196
4	36	25221
6	36	24596
… …	… …	… …

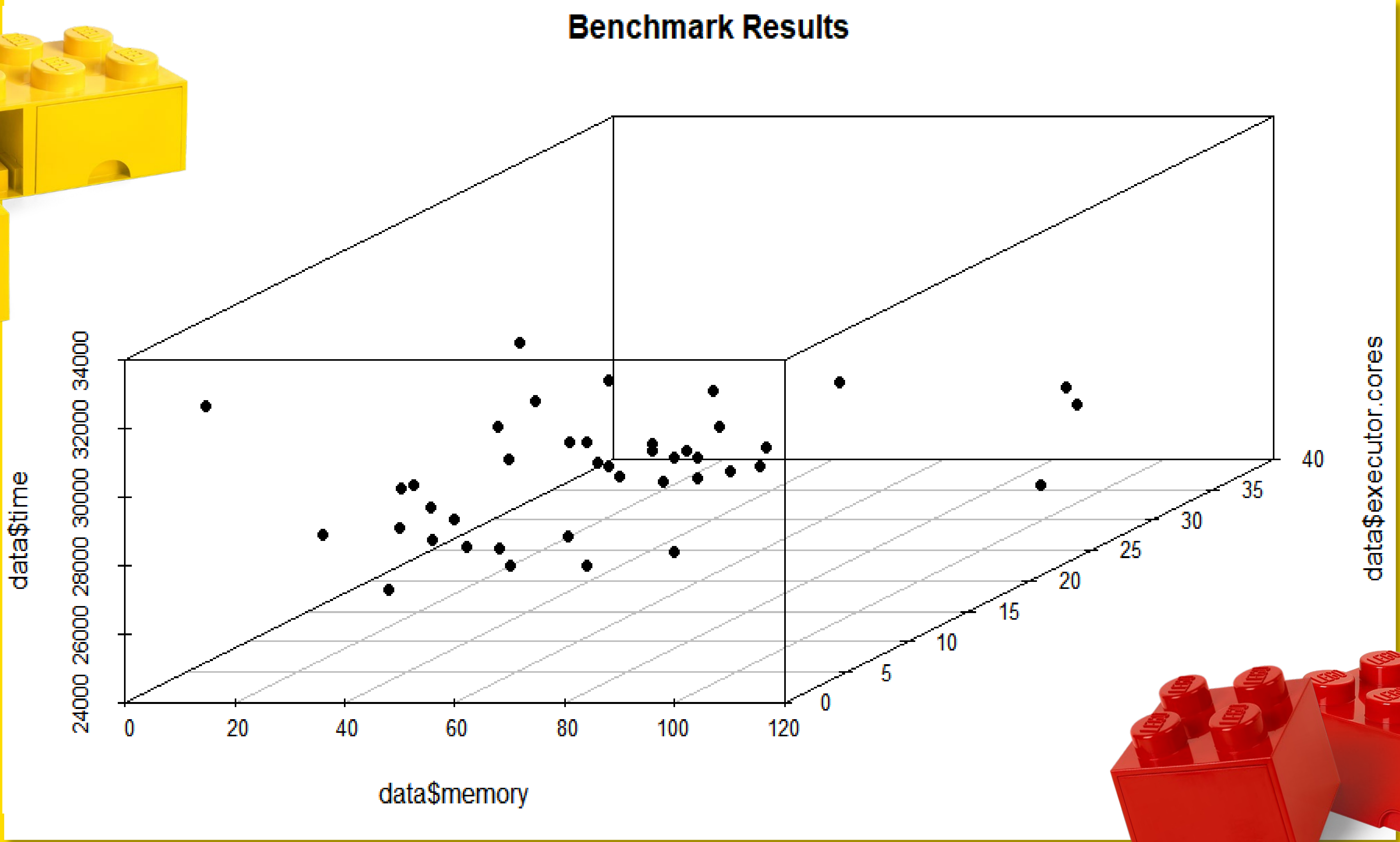


```
print(conf.toDebugString())
conf = SparkConf() . setAll([('spark.executor.memory','8g'),('spark.executor.cores','3'),('spark.cores.max','3'),('spark.driver.memory','8g')])
#conf.set('spark.dynamicAllocation.enabled','True')
```



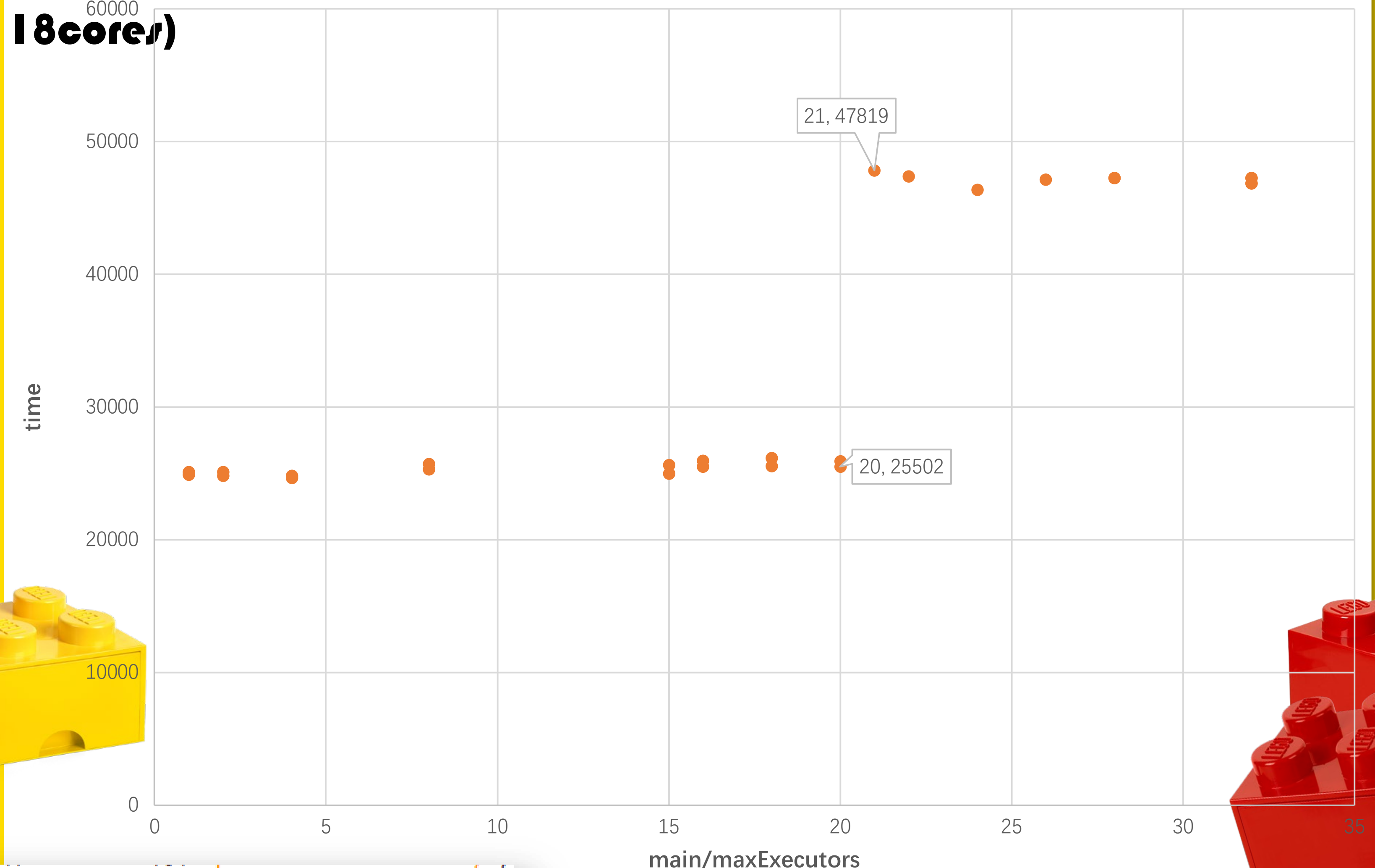


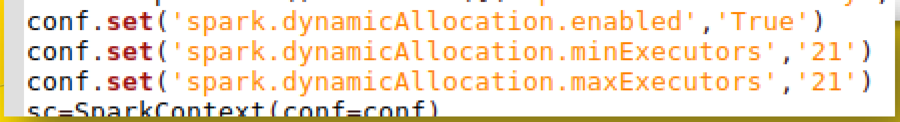
Benchmark Results



6.minExecutors & maxExecutors configuration

(with 64GB RAM, 18cores)





```
conf.set('spark.dynamicAllocation.enabled','True')
conf.set('spark.dynamicAllocation.minExecutors','21')
conf.set('spark.dynamicAllocation.maxExecutors','21')
sc=SparkContext(conf=conf)
```

Thanks for your attention.