

LSDM-Project-pysprak

March 29, 2022

```
[1]: pip install pyspark
```

```
Collecting pyspark
  Downloading pyspark-3.2.1.tar.gz (281.4 MB)
    |                                     | 281.4 MB 34 kB/s
Collecting py4j==0.10.9.3
  Downloading py4j-0.10.9.3-py2.py3-none-any.whl (198 kB)
    |                                     | 198 kB 59.8 MB/s
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.2.1-py2.py3-none-any.whl
size=281853642
sha256=d8213f9983b5b4964d37a870adf62808f9aaa8aa34c79db9c29bd70c67dada01
  Stored in directory: /root/.cache/pip/wheels/9f/f5/07/7cd8017084dce4e93e84e92e
fd1e1d5334db05f2e83bcef74f
Successfully built pyspark
Installing collected packages: py4j, pyspark
Successfully installed py4j-0.10.9.3 pyspark-3.2.1
```

```
[2]: pip install findspark
```

```
Collecting findspark
  Downloading findspark-2.0.1-py2.py3-none-any.whl (4.4 kB)
Installing collected packages: findspark
Successfully installed findspark-2.0.1
```

0.1 Importing libraries and modules

```
[3]: import pyspark
from pyspark.sql import *
from pyspark.sql.types import *
from pyspark import SparkContext, SparkConf
from pyspark.sql import *
import pyspark.sql.functions as F
from pyspark.sql.functions import col, udf
from pyspark.sql.types import *
from pyspark.sql import types as T
```

```
import findspark
```

```
[4]: findspark.init()
```

0.2 Create a Spark session named zillow and construct a dataframe in order to load our data. Providing also a schema for our data

```
[5]: spark = SparkSession.builder.appName("zillow").getOrCreate()
```

```
[6]: df = spark.read.format('csv').option('header', True).load('zillow.csv')
```

```
[7]: df.printSchema()
```

```
root
 |-- title: string (nullable = true)
 |-- address: string (nullable = true)
 |-- city: string (nullable = true)
 |-- state: string (nullable = true)
 |-- postal_code: string (nullable = true)
 |-- price: string (nullable = true)
 |-- facts and features: string (nullable = true)
 |-- real estate provider: string (nullable = true)
 |-- url: string (nullable = true)
```

```
[8]: df.show(10)
```

```
+-----+-----+-----+-----+-----+-----+-----+
|      title|address|      city|state|postal_code|      price| facts and
features|real estate provider|      url|
+-----+-----+-----+-----+-----+-----+-----+
|Condo for sale| null| Somerville| MA|      02145| $342,000|2 bds, 1.0 ba
,70...|William Raveis R...|https://www.zillo...|
|Condo for sale| null|      Boston| MA|      02116|$1,700,000|2 bds, 2.0 ba
,12...|Century 21 North ...|https://www.zillo...|
|Condo for sale| null|      Boston| MA|      02118| $336,500|1 bds, 1.0 ba
,10...|Maloney Propertie...|https://www.zillo...|
|House for sale| null|      Boston| MA|      02118|$9,950,000|4 bds, 7.0 ba
,68...|Campion & Company...|https://www.zillo...|
|Condo for sale| null|      Boston| MA|      02128| $479,000|2 bds, 3.0 ba
,10...|Berkshire Hathawa...|https://www.zillo...|
|House for sale| null| East Boston| MA|      02128| $899,000|3 bds, 3.0 ba
,23...|Berkshire Hathawa...|https://www.zillo...|
```

```
|Condo for sale| null| Somerville| MA| 02145| $397,300|2 bds, 1.0 ba
,78...|William Raveis R...|https://www.zillo...|
|Condo for sale| null|South Boston| MA| 02127| $619,900|2 bds, 1.0 ba
,85...|Bento Real Estate...|https://www.zillo...|
|Condo for sale| null| Boston| MA| 02116| $850,000|1 bds, 1.0 ba
,67...|Engel & Volkers B...|https://www.zillo...|
|Condo for sale| null| Boston| MA| 02114| $649,900|2 bds, 1.0 ba
,51...|Leading Edge Real...|https://www.zillo...|
+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+
only showing top 10 rows
```

0.3 Task 1:

For the first task we have to retrieve the number of bedrooms from the column of ‘facts and features’. So to solve it we construct a function first. At the first step, we use split with “,” and taking the first column [0], getting the value f.i. ‘2 bds’, after we use split again with ” ” as definition and take the first column again [0] in order to get finally only the numbers f.i. ‘2’. In the sequel, in order to face the “None” values, because some listings contain them, i decide to replace them with “0”. Afterwards, constructing a udf function with name “convertUDF” we apply it on the column of ‘fact and features’, receiving this way the number of bedrooms. At the last part of the processing we add a new column in the same dataframe, with the filtered values in order to use the cleaned column later. Moreover we present for the first 10 listings the corresponding bedrooms and later we provide the distinct values for the column to verify the above processing.

```
[9]: def retrieve_bds(string):
      string = string.split(",")[0]
      string = string.split(" ")[0]
      if (string == 'None'): # replace the None values with 0
          string = "0"
      return int(string) # returning as integer each value

      convertUDF = udf(lambda z: retrieve_bds(z), IntegerType()) # specify the type of
      ↪ column
      df = df.withColumn('bedrooms', convertUDF(col('facts and features')))
```

```
[10]: df.select('bedrooms').show(10)
```

```
+-----+
|bedrooms|
+-----+
|      2|
|      2|
|      1|
|      4|
```

	2
	3
	2
	2
	1
	2

+-----+

only showing top 10 rows

```
[11]: df.select('bedrooms').distinct().show()
```

+-----+

bedrooms

+-----+

	12
	1
	13
	6
	16
	3
	5
	15
	9
	17
	4
	8
	7
	10
	11
	14
	2
	0
	18

+-----+

0.4 Task 2:

The rationale for this task is the same as above. This time i split first with “,” and with column [1] i took the values f.i. ” 1.0 ba”, then i split again with ” ” and choose the column [1] to get the values f.i “1.0”, because there was a space at the start of ” 1.0 ba”. Finally with the last split on “.” and choosing the column [0] i took only the number of bathrooms. The following steps is the same described above with more details.

```
[12]: def retrieve_ba(string):
        string = string.split(",")[1].split(' ')[1]
        string=string.split(".")[0]
        if (string == 'None'): # replace the None with 0
            string = "0"
        return int(string)

        convertUDF = udf(lambda z: retrieve_ba(z), IntegerType())
        df = df.withColumn('bathrooms', convertUDF(col('facts and features')))
```

```
[13]: df.select('bathrooms').show(10)
```

```
+-----+
|bathrooms|
+-----+
|         1|
|         2|
|         1|
|         7|
|         3|
|         3|
|         1|
|         1|
|         1|
|         1|
+-----+
```

only showing top 10 rows

```
[14]: df.select('bathrooms').distinct().show()
```

```
+-----+
|bathrooms|
+-----+
|        12|
|         1|
|        13|
|         6|
|         3|
|         5|
|         9|
|         4|
|         8|
|         7|
|        10|
|        11|
|         2|
|         0|
```

```
+-----+
```

0.5 Task 3:

The rationale for this task is the same as above. This time i split first with “,” and with column [2] i took the values f.i. “705 sqft”, then i split again with ” ” and choose the column [0] to get the values f.i. “705” which is the number of sqft. The following steps is the same described above with more details.

```
[15]: def retrieve_sqft(string):
      string = string.split(',')[2].split(' ')[0]
      return int(string)

convertUDF = udf(lambda z: retrieve_sqft(z), IntegerType())
df = df.withColumn('sqft', convertUDF(col('facts and features')))
```

```
[16]: df.select('sqft').show(10)
```

```
+-----+
```

```
|sqft|
```

```
+-----+
```

```
| 705|
```

```
|1228|
```

```
|1000|
```

```
|6836|
```

```
|1000|
```

```
|2313|
```

```
| 780|
```

```
| 856|
```

```
| 675|
```

```
| 511|
```

```
+-----+
```

only showing top 10 rows

0.6 Task 4:

For this task i create a list with the accepted values in order to filter the column of ‘title’. As accepted categories i will qulify the ‘Condo’, ‘House’, ‘Lot’ and ‘Uknown-Type’ for the rest instances. So first i split the column of ‘title’ which contain f.i. “Condo for sale” and get the column [0] to recieve f.i. the “Condo”. Afterwards, constructing the above list i filter every listing with the accepted list and return the corresponding value unless the values does not appear on the list, these cases masked as UknownType. The following steps is the same described above with more details.

```
[17]: def retrieve_type(string):

    string=string.split()[0]
    accepted_list=["Condo", "House","Lot" ]
    if string not in accepted_list:
        return "UknownType"
    else:
        return string

convertUDF = udf(lambda z: retrieve_type(z))#default type is string
df = df.withColumn('type', convertUDF(col('title')))
```

```
[18]: df.select('type').show(10)
```

```
+-----+
| type|
+-----+
|Condo|
|Condo|
|Condo|
|House|
|Condo|
|House|
|Condo|
|Condo|
|Condo|
+-----+
only showing top 10 rows
```

```
[19]: df.select('type').distinct().show()
```

```
+-----+
|      type|
+-----+
|      Lot|
|      Condo|
|      House|
|UknownType|
+-----+
```

0.7 Task 5:

At this part for each listing f.i. “Condo for sale” on the ‘title’ column, i split first with ” ” and with [-1] i took f.i the “sale” i.e. the ending. There are not listings ‘rent’ or

‘sold’ so the qualified categories are ‘foreclose’ and ‘sale’. I took the assumption, that the listing with “New construction” as value, implies they are for sale, for this reason i report them as listings for sale.

```
[20]: def retrieve_offer(string):
      string=string.split()[-1]# nothing defined on split has default to split on "
      ↪"
      if string == "construction":
          return "sale"
      elif string == "Foreclosure":
          return "foreclose"
      else:
          return string

      convertUDF = udf(lambda z: retrieve_offer(z))
      df = df.withColumn('offer', convertUDF(col('title')))
```

```
[21]: df.select('offer').show(10)
```

```
+-----+
|offer|
+-----+
| sale|
| sale|
| sale|
| sale|
| sale|
| sale|
| sale|
| sale|
| sale|
| sale|
| sale|
+-----+
only showing top 10 rows
```

```
[22]: df.select('offer').distinct().show()
```

```
+-----+
|  offer|
+-----+
|foreclose|
|   sale|
+-----+
```

0.8 Task 6:

For this part, in order to filter out the listings that are not for sale, i use the function 'contains' and passing the 'sale' as definition i took only the listings that are for sale, hence i filter out the listings that are not for sale. Mentioning here, that i use the 'offer' column, which contains the trasnformed values from task 5 containing the listings that are for "sale" or "forclose". Finally i provide some statistics, for the number of listings overall, the number of listings which are for sale, and how many listings are not for sale.

```
[23]: df_listings_for_sale = df.filter(df.offer.contains('sale'))
df_listings_for_sale.
→select('title','address','city','state','postal_code','price','facts and
→features','real estate provider','url').show(10)
```

```
+-----+-----+-----+-----+-----+-----+-----+
|      title|address|      city|state|postal_code|      price|  facts and
features|real estate provider|      url|
+-----+-----+-----+-----+-----+-----+-----+
|Condo for sale|  null|  Somerville|  MA|      02145|  $342,000|2 bds, 1.0 ba
,70...|William Raveis R...|https://www.zillo...|
|Condo for sale|  null|      Boston|  MA|      02116|$1,700,000|2 bds, 2.0 ba
,12...|Century 21 North ...|https://www.zillo...|
|Condo for sale|  null|      Boston|  MA|      02118|  $336,500|1 bds, 1.0 ba
,10...|Maloney Propertie...|https://www.zillo...|
|House for sale|  null|      Boston|  MA|      02118|$9,950,000|4 bds, 7.0 ba
,68...|Campion & Company...|https://www.zillo...|
|Condo for sale|  null|      Boston|  MA|      02128|  $479,000|2 bds, 3.0 ba
,10...|Berkshire Hathawa...|https://www.zillo...|
|House for sale|  null|  East Boston|  MA|      02128|  $899,000|3 bds, 3.0 ba
,23...|Berkshire Hathawa...|https://www.zillo...|
|Condo for sale|  null|  Somerville|  MA|      02145|  $397,300|2 bds, 1.0 ba
,78...|William Raveis R...|https://www.zillo...|
|Condo for sale|  null|South Boston|  MA|      02127|  $619,900|2 bds, 1.0 ba
,85...|Bento Real Estate...|https://www.zillo...|
|Condo for sale|  null|      Boston|  MA|      02116|  $850,000|1 bds, 1.0 ba
,67...|Engel & Volkers B...|https://www.zillo...|
|Condo for sale|  null|      Boston|  MA|      02114|  $649,900|2 bds, 1.0 ba
,51...|Leading Edge Real...|https://www.zillo...|
+-----+-----+-----+-----+-----+-----+-----+
only showing top 10 rows
```

```
[24]: print(f"the number of listings overall is :{df.count()}")
print(f"the number of listings for sale is :{df_listings_for_sale.count()}")
print(f"they have been filtered: {df.count()-df_listings_for_sale.count()}")
→listings (number of listings not for sale)")
```

the number of listings overall is :100313
the number of listings for sale is :100018
they have been filtered: 295 listings (number of listings not for sale)

0.9 Task 7:

At this part, for each listing f.i. “\$342,000” in order to receive only the number without the symbols i did the following steps. First choosing [1:] for each row i avoid the “\$”. Secondly, using the function ‘filter’ i check the given string to be digit and with the ‘join’ function later i succeed to concatenate the values before and after the “,” without taking into account the “.”.

```
[25]: def retrieve_price(string):  
    price = string[1:] #avoid $ from each listing  
    #check each listing for digit and with join avoid the '.'  
    numeric_filter = filter(str.isdigit, price)  
    numeric_string = "".join(numeric_filter)  
    return int(numeric_string)  
  
convertUDF = udf(lambda z: retrieve_price(z), IntegerType())  
df = df.withColumn('price_filtered', convertUDF(col('price')))
```

```
[26]: df.select(convertUDF(col('price')).alias('price_filtered')).show(10)
```

```
+-----+  
|price_filtered|  
+-----+  
|      342000|  
|     1700000|  
|      336500|  
|     9950000|  
|      479000|  
|      899000|  
|      397300|  
|      619900|  
|      850000|  
|      649900|  
+-----+  
only showing top 10 rows
```

0.10 Task 8:

The rationale for this task is similar with the task 6. I use the column of “bedrooms” constrasting on task 1 which contain the number of bedrooms. So using the ‘filter’ function i took the listings with less than 11 bedrooms. Hence i filter out the listings that have more than 10 bedrooms. Finally i provide some statistics, for the number

of listings overall, the number of listings with less than 10 bedrooms, and how many listings are filtered out (listings with more than 10 bedrooms).

```
[27]: data_retrieve_bds_more_than_ten = df.filter((df.bedrooms < 11))
data_retrieve_bds_more_than_ten.
↳select('title','address','city','state','postal_code','price','facts and_
↳features','real estate provider','url').show(10)
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
|      title|address|      city|state|postal_code|      price| facts and
features|real estate provider|      url|
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
|Condo for sale| null| Somerville| MA|      02145| $342,000|2 bds, 1.0 ba
,70...|William Raveis R...|https://www.zillo...|
|Condo for sale| null| Boston| MA|      02116|$1,700,000|2 bds, 2.0 ba
,12...|Century 21 North ...|https://www.zillo...|
|Condo for sale| null| Boston| MA|      02118| $336,500|1 bds, 1.0 ba
,10...|Maloney Propertie...|https://www.zillo...|
|House for sale| null| Boston| MA|      02118|$9,950,000|4 bds, 7.0 ba
,68...|Campion & Company...|https://www.zillo...|
|Condo for sale| null| Boston| MA|      02128| $479,000|2 bds, 3.0 ba
,10...|Berkshire Hathawa...|https://www.zillo...|
|House for sale| null| East Boston| MA|      02128| $899,000|3 bds, 3.0 ba
,23...|Berkshire Hathawa...|https://www.zillo...|
|Condo for sale| null| Somerville| MA|      02145| $397,300|2 bds, 1.0 ba
,78...|William Raveis R...|https://www.zillo...|
|Condo for sale| null|South Boston| MA|      02127| $619,900|2 bds, 1.0 ba
,85...|Bento Real Estate...|https://www.zillo...|
|Condo for sale| null| Boston| MA|      02116| $850,000|1 bds, 1.0 ba
,67...|Engel & Volkers B...|https://www.zillo...|
|Condo for sale| null| Boston| MA|      02114| $649,900|2 bds, 1.0 ba
,51...|Leading Edge Real...|https://www.zillo...|
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
```

only showing top 10 rows

```
[28]: print(f"the number of listings overall is :{df.count()}")
print(f"the number of listings with less than 10 bedrooms :
↳{data_retrieve_bds_more_than_ten.count()}")
print(f"they have been filtered: {df.count()-data_retrieve_bds_more_than_ten.
↳count()} listings" )
```

```
the number of listings overall is :100313
the number of listings with less than 10 bedrooms :99663
they have been filtered: 650 listings
```

0.11 Task 9:

The rationale for this task is similar with the task 6 and 8. I use the column of “price_filtered” constructed on task 7 which contain the prices. So using the ‘filter’ function i took the listings with less than 20000000 price and more than 100000. Hence, i filter out the listings with more than 20000000 price and less than 100000 price. Finally i provide some statistics, for the number of listings overall, the number of listings with less than 20000000 price and more than 100000, and how many listings are filtered out (listings with more than 20000000 price and less than 100000 price).

```
[29]: df_listings_most_expensive_cheapest = df.filter((df.price_filtered < 20000000)
↳ & (df.price_filtered > 100000))
df_listings_most_expensive_cheapest.
↳ select('title', 'address', 'city', 'state', 'postal_code', 'price', 'facts and
↳ features', 'real estate provider', 'url').show(10)
```

```
+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
|      title|address|      city|state|postal_code|      price| facts and
features|real estate provider|      url|
+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
|Condo for sale|  null| Somerville|  MA|      02145|  $342,000|2 bds, 1.0 ba
,70...|William Raveis R...|https://www.zillo...|
|Condo for sale|  null|      Boston|  MA|      02116|$1,700,000|2 bds, 2.0 ba
,12...|Century 21 North ...|https://www.zillo...|
|Condo for sale|  null|      Boston|  MA|      02118|  $336,500|1 bds, 1.0 ba
,10...|Maloney Propertie...|https://www.zillo...|
|House for sale|  null|      Boston|  MA|      02118|$9,950,000|4 bds, 7.0 ba
,68...|Campion & Company...|https://www.zillo...|
|Condo for sale|  null|      Boston|  MA|      02128|  $479,000|2 bds, 3.0 ba
,10...|Berkshire Hathawa...|https://www.zillo...|
|House for sale|  null| East Boston|  MA|      02128|  $899,000|3 bds, 3.0 ba
,23...|Berkshire Hathawa...|https://www.zillo...|
|Condo for sale|  null| Somerville|  MA|      02145|  $397,300|2 bds, 1.0 ba
,78...|William Raveis R...|https://www.zillo...|
|Condo for sale|  null|South Boston|  MA|      02127|  $619,900|2 bds, 1.0 ba
,85...|Bento Real Estate...|https://www.zillo...|
|Condo for sale|  null|      Boston|  MA|      02116|  $850,000|1 bds, 1.0 ba
,67...|Engel & Volkers B...|https://www.zillo...|
|Condo for sale|  null|      Boston|  MA|      02114|  $649,900|2 bds, 1.0 ba
,51...|Leading Edge Real...|https://www.zillo...|
+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
only showing top 10 rows
```

```
[30]: print(f"the number of listings overall is :{df.count()}")
      print(f"the number of listings with less than 20000000$ and more than 100000$ :
      ↳{df_listings_most_expensive_cheapest.count()}")
      print(f"they have been filtered: {df.
      ↳count()-df_listings_most_expensive_cheapest.count()} listings" )
```

```
the number of listings overall is :100313
the number of listings with less than 20000000$ and more than 100000$ :98838
they have been filtered: 1475 listings
```

0.12 Task 10:

At this part of the exercise i use the column of 'type' which constructed on task 4, passing in the 'filter' function the 'House' i receive only the listings which are Houses. Hence i filter out all the listings that are not houses. Finally, i provide some statistics, for the number of listings overall, the number of listings which are Houses, and how many listings are filtered out (listings which are not houses).

```
[31]: df_listings_that_are_houses = df.filter(df.type=="House")
      df_listings_that_are_houses.
      ↳select('title','address','city','state','postal_code','price','facts and
      ↳features','real estate provider','url').show(10)
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
|      title|address|      city|state|postal_code|      price| facts and
features|real estate provider|      url|
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
|House for sale| null|      Boston|  MA|      02118|$9,950,000|4 bds, 7.0 ba
,68...|Campion & Company...|https://www.zillo...|
|House for sale| null| East Boston|  MA|      02128|  $899,000|3 bds, 3.0 ba
,23...|Berkshire Hathawa...|https://www.zillo...|
|House for sale| null|      Boston|  MA|      02113|$1,200,000|2 bds, 1.0 ba
,11...|      CL Properties|https://www.zillo...|
|House for sale| null|      Boston|  MA|      02129|$1,119,000|3 bds, 4.0 ba
,16...|All Star Realty, ...|https://www.zillo...|
|House for sale| null|South Boston|  MA|      02127|$1,699,000|4 bds, 3.0 ba
,20...|      Compass|https://www.zillo...|
|House for sale| null|      Boston|  MA|      02128|  $589,000|2 bds, 2.0 ba
,12...|Gibson Sotheby's ...|https://www.zillo...|
|House for sale| null|      Boston|  MA|      02115|$9,750,000|4 bds, 7.0 ba
,47...|MGS Group Real Es...|https://www.zillo...|
|House for sale| null| Somerville|  MA|      02145|$2,075,000|4 bds, 4.0 ba
,31...|Gibson Sotheby's ...|https://www.zillo...|
|House for sale| null|      Boston|  MA|      02114|$3,200,000|3 bds, 3.0 ba
,20...|William Raveis R...|https://www.zillo...|
```

```
|House for sale| null|South Boston| MA| 02127|$1,175,000|3 bds, 3.0 ba
,18...|Coldwell Banker R...|https://www.zillo...|
```

```
+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+-----+-----+
```

only showing top 10 rows

```
[32]: print(f"the number of listings overall is :{df.count()}")
      print(f"the number of listings which are houses:{df_listings_that_are_houses.
      ↪count()}")
      print(f"they have been filtered:{df.count()-df_listings_that_are_houses.
      ↪count()},listings" )
```

```
the number of listings overall is :100313
the number of listings which are houses:22554
they have been filtered:77759,listings
```

0.13 Task 11:

At the final task of the exercise, i use the 'sql' function passing a query to get the average price per sqft for houses that are for sale and group them by bedrooms. To begin with we construct a 'View' named 'task11' in which the sql function could processing, containing all the dataset including the extra columns constructed from the previous tasks. Afterwards, inside the statement we start by selecting the bedrooms which is a column constructed in the first task. We take the avg() of each listing from "price_filtered" column constructed in the task 7 divided of each listing from "sqft" column constructed in task 3, getting this way the average price per sqft. We specify in where statement that, from the 'title' column we keeping only the listings that are 'House for sale'. Finally we group them by the number of bedrooms.

```
[33]: df.createOrReplaceTempView("task11")
```

```
[35]: spark.sql("select bedrooms,avg(price_filtered/sqft) as average_price_per_sqft
      ↪from task11 where title == 'House for sale' group by bedrooms order by
      ↪bedrooms").show()
```

```
+-----+-----+
|bedrooms|average_price_per_sqft|
+-----+-----+
|      0|          1250.0|
|      2|  716.0381965996851|
|      3|  678.9521125584703|
|      4|   909.14739964404|
|      5|  908.8325677804108|
|      6|  422.3111656297121|
|      7| 1126.0252348993295|
|      8| 1567.6470588235286|
|      9| 1108.1412183984862|
```

| 11 | 433.6545589325418 |
+-----+-----+