Practice with PySpark

ALY 6110 Data Management & Big Data

**CHIRAG SANJAYBHAI SHAH**

Northeastern University



**Note**:

Chirag Sanjaybhai Shah College of Professional Studies, Northeastern University, Boston, MA 02115.

This report was created as a part of the project to gain hands-on experience in PySpark.

**Instructor Name: Janos Mako**

Chirag Sanjaybhai Shah is now a student at Department of Analytics, Northeastern University

Contact:

[shah.chir@husky.neu.edu](mailto:shah.chir@husky.neu.edu)

**INTRODUCTION**

We have given the two datasets to get some hands on in PySpark. Previously we installed the pyspark in our system while doing Lab2 assignment. To proceed further, we have been provided with two datasets. The PySpark is useful in to do big data analysis. PySpark supports SQL queries and helps the user to get precise information. PySpark has in memory processing which helps in doing the analyzing the queries at a fast rate. PySpark has the characteristic of fault tolerance capacity. This comes from the resilient distributed dataset (RDD) in PySpark. RDD can supports various types of languages such as Python, Java, Scala, R, etc.

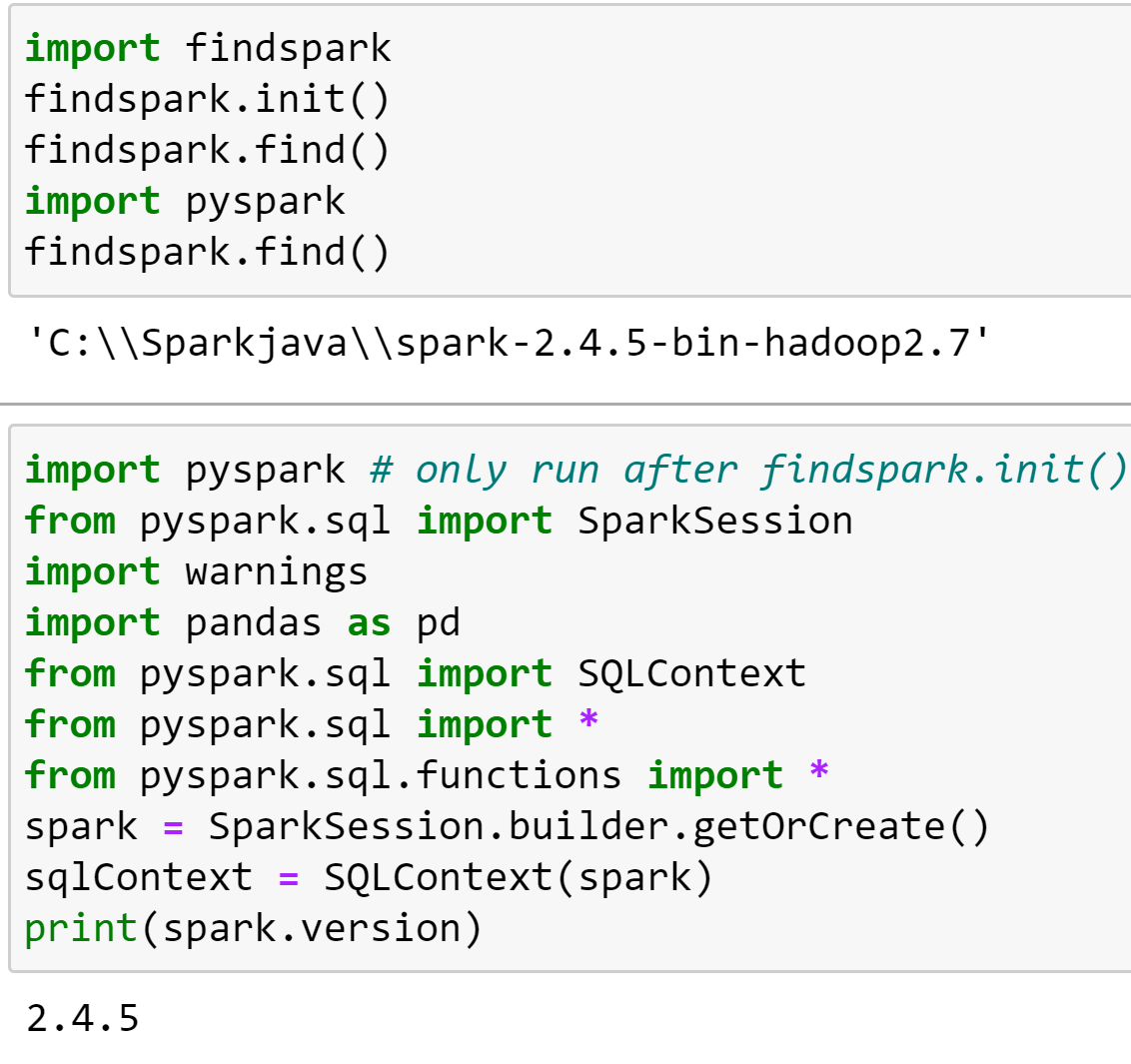
First dataset has been provided with housing data. The Zillo Home Value Index (ZHVI) estimates of the each region are given here. The ZHVI helps the buyer to understand the value of the each particular housing. We have been provided with 19 variables. The dataset gives the information of when the ZHVI is highest at month, quarter, year for the particular region. The dataset is provided for United States regions. The Zip\_Zhvi\_Summary\_AllHomes consist of approx. 30k rows.

Second dataset has been provided with housing data of House Price Index (HPI) value. The HPI values are measured to let person know the movement of prices of the single family. The HPI is used to measure the house prices trends. The HPI is measured from the specific date. Here, in this dataset we have been provided the HPI values for many zip codes of United States. Zip codes given are with 3 letters. To compare the HPI values we are also being provided with HPI values of the years 1990 and 2000. The dataset HPI\_AT\_BDL\_ZIP3 contains aprrox 35k rows with 6 variables.

For both the datasets, we have done the exploratory analysis and find the correlations of the data. This is done with the help of PySpark commands.

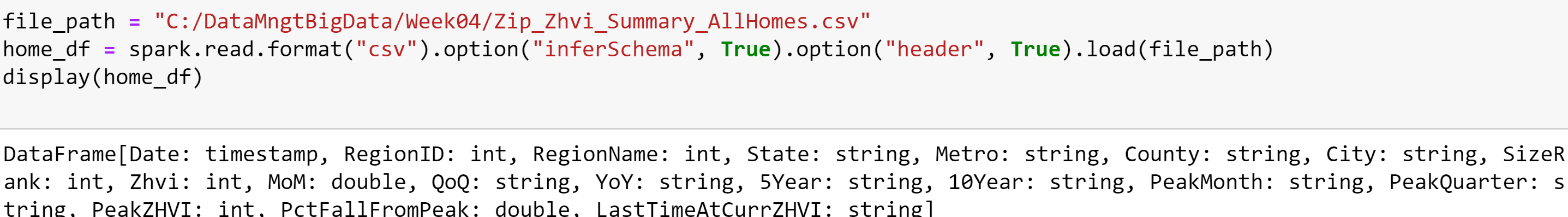
**ANALYSIS**

Importing the essential libraries: PySpark libraries are imported to perform the pyspark commands by running SQL queries. Python libraries are imported to convert the pypsark dataframe to pandas dataframe and do the data visualization.

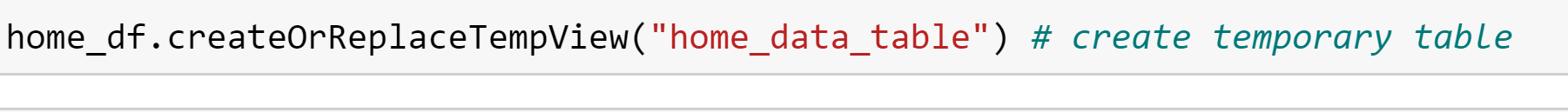


**PART-1: Zip\_Zhvi\_Summary\_AllHomes Analysis**

1. Importing dataset:The dataset is being loaded in pyspark dataframe.



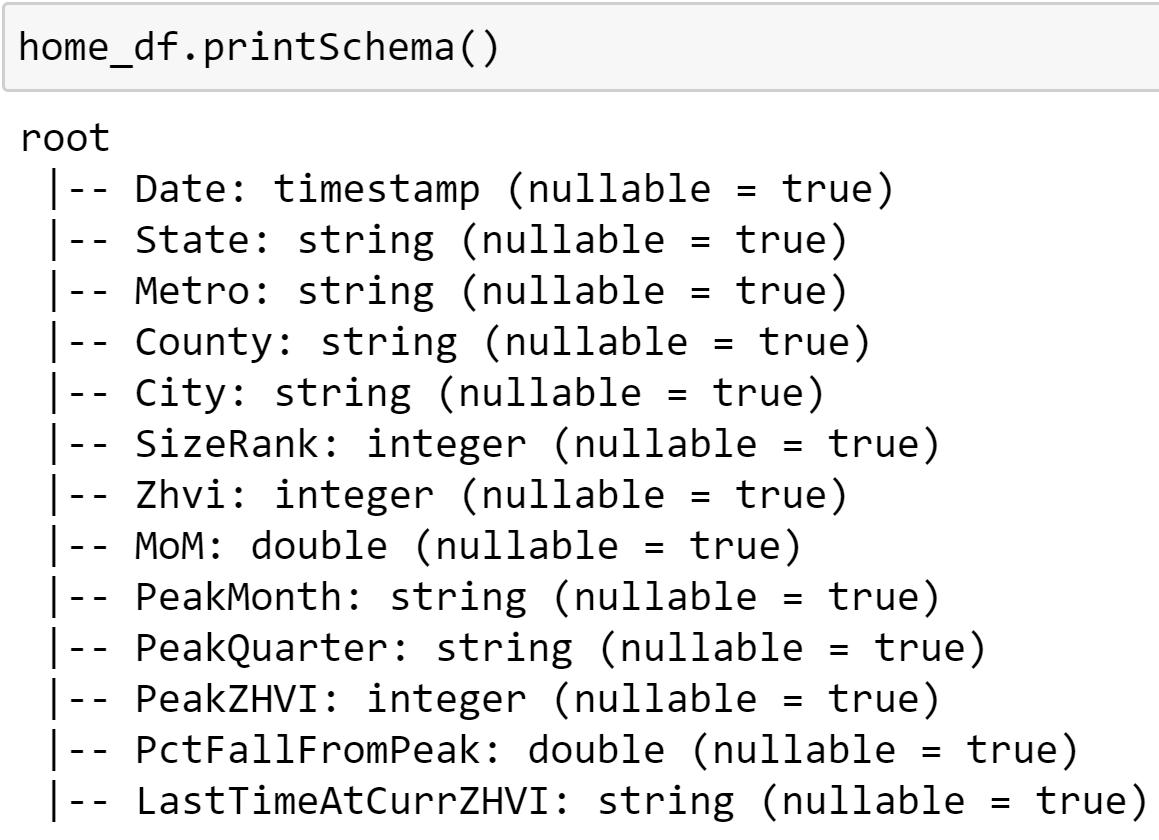
1. Temporary Table: We made the spark temporary table to run the SQL queries.



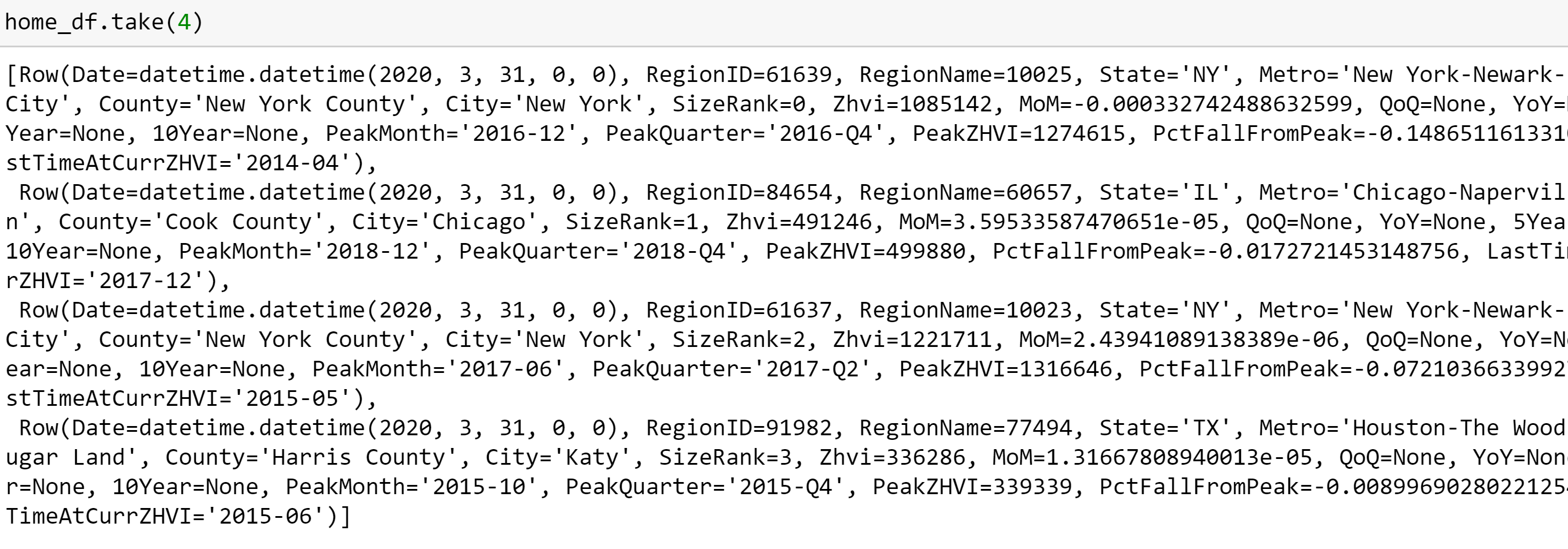
1. Columns drop: We dropped the unnecessary columns as we have the NULL values and no entry present in those columns:



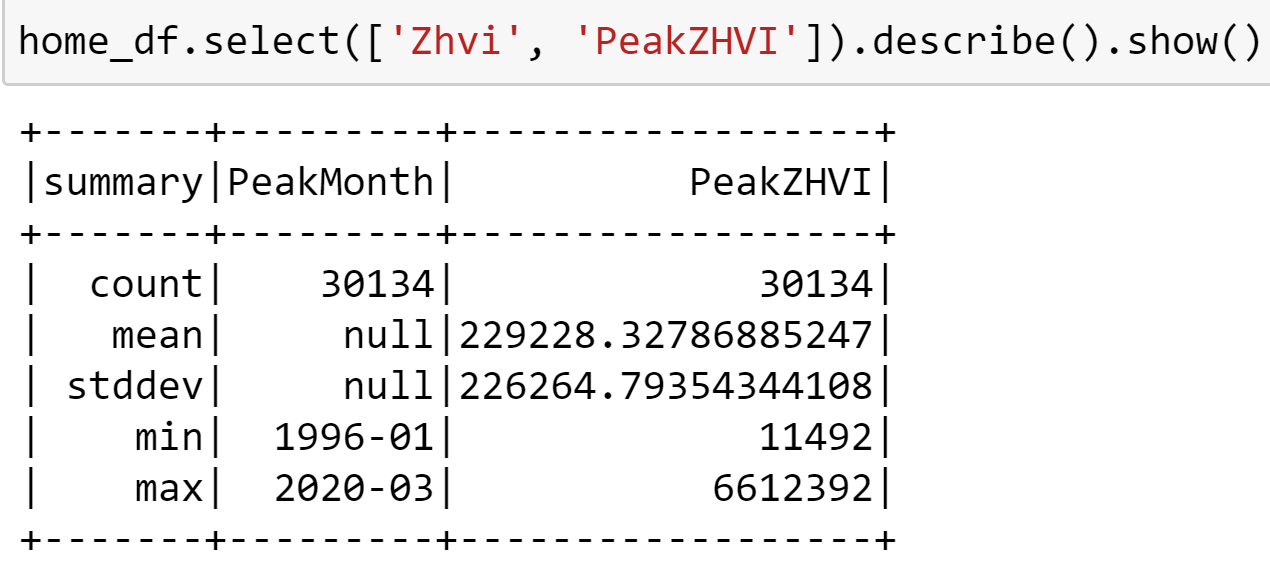
1. Schema: The printSchema function helps to view the type of coulmns in the table:



1. Displaying some rows of the table: We have used take() to display the rows of the table:



1. Information of the columns: We used describe() to show the range of the values for ZHVI columns:



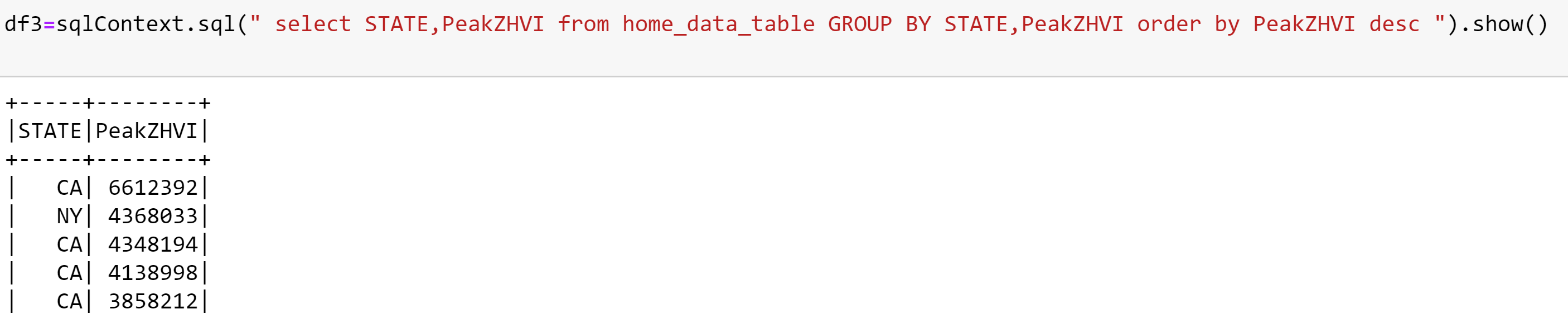
1. Conversion to Pandas data frame:



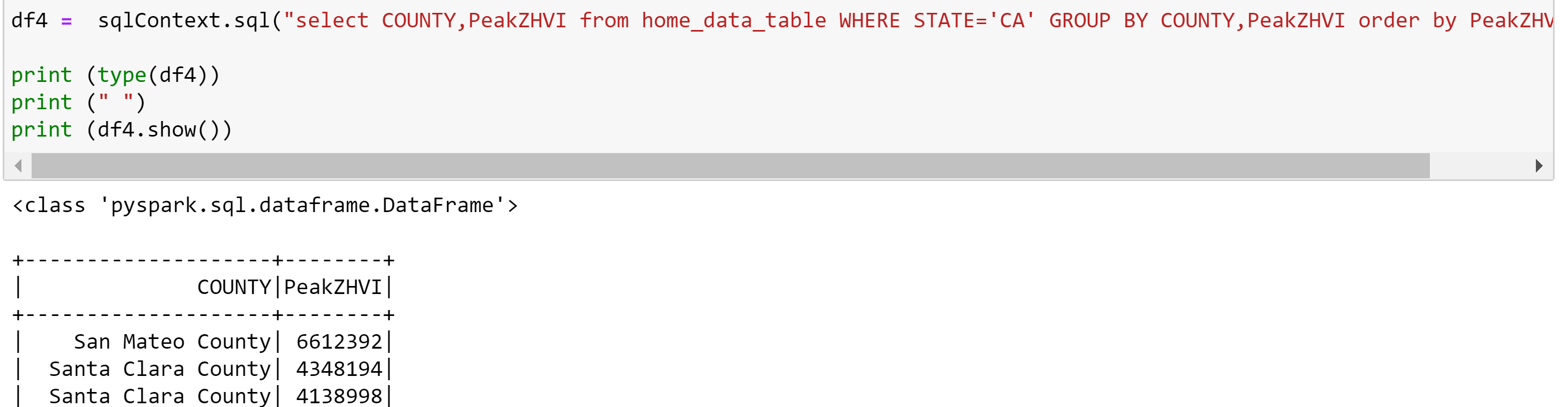
1. Distribution of the ZHVI and PeakZHVI values:



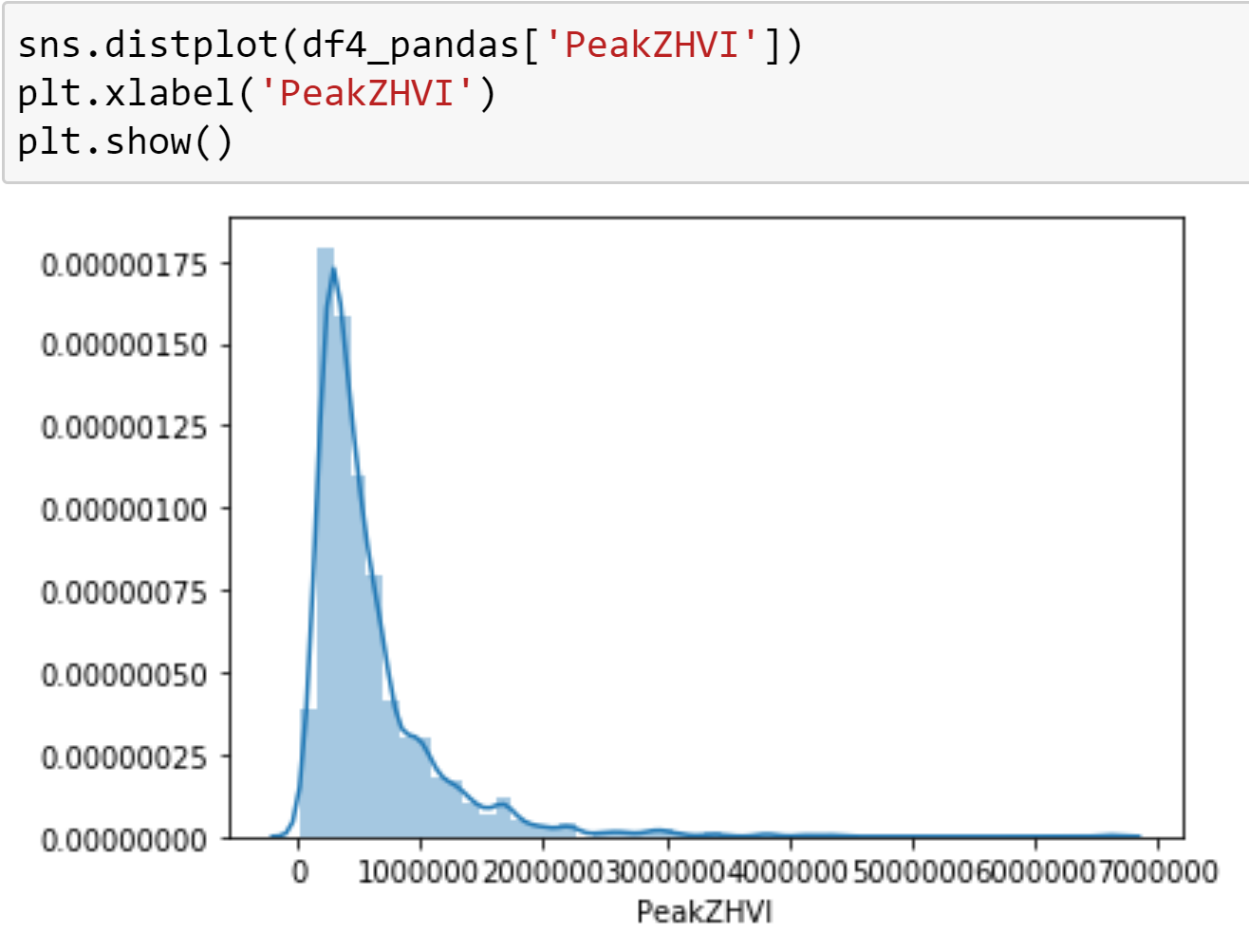
1. From the below query we can say that CA has the highest PeakZHVI values:



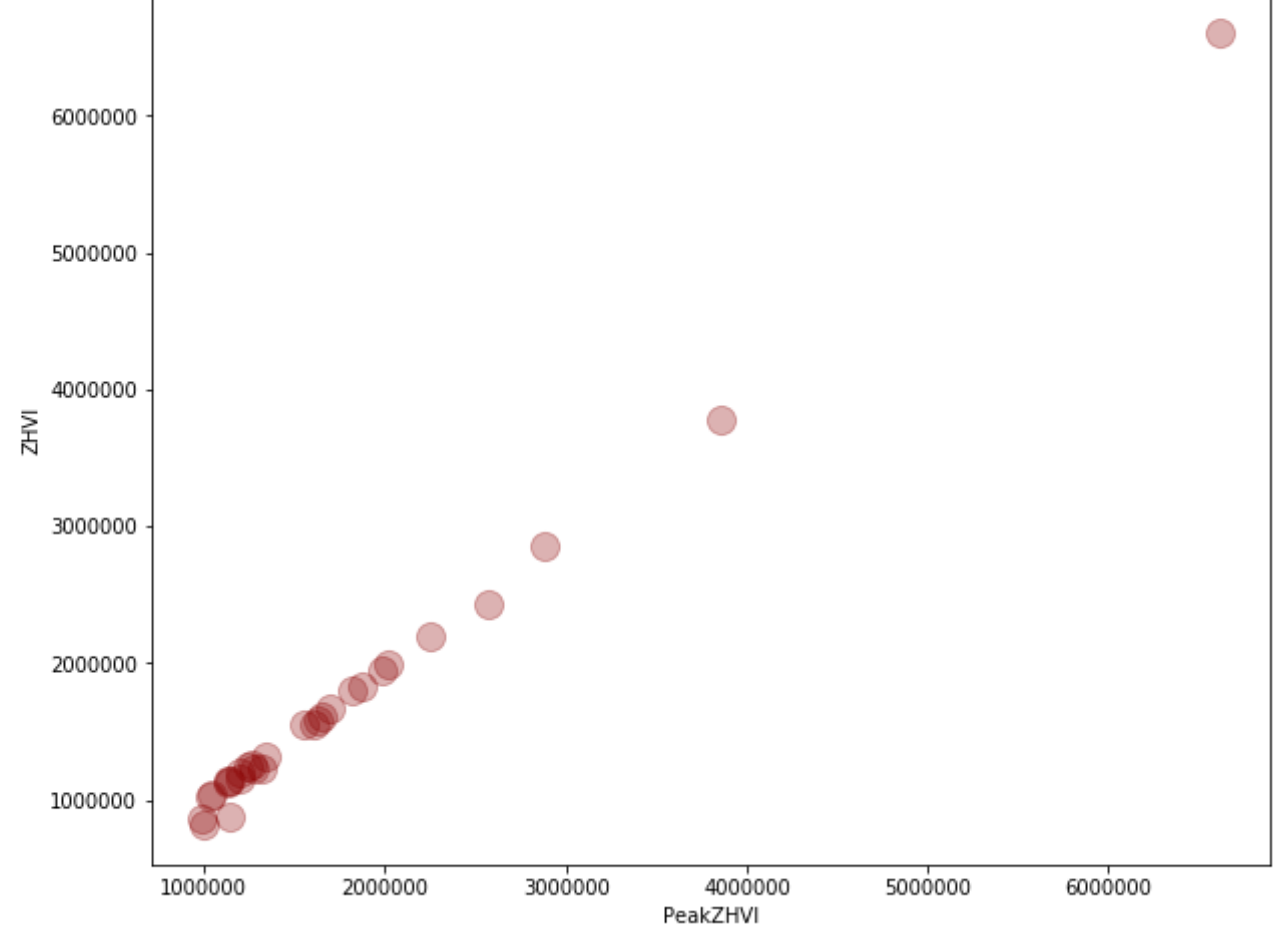
1. We digged deeper to see that which CA county has the highest PeakZHVI: From the below query we see that San Mateo County has the highest value.



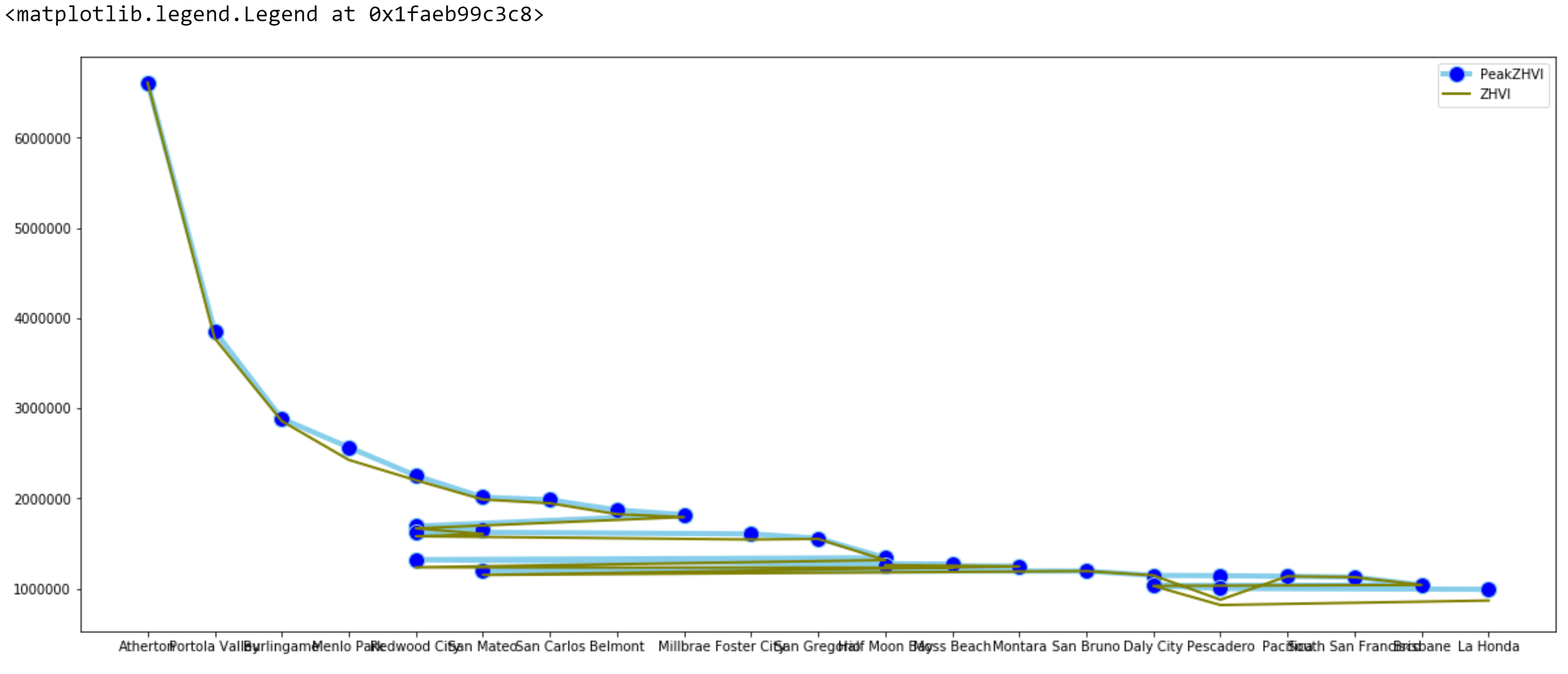
1. Distribution of PeakZHVI values for San Mateo County county:



1. Distribution of PeakZHVI and ZHVI values for each county in CA state:

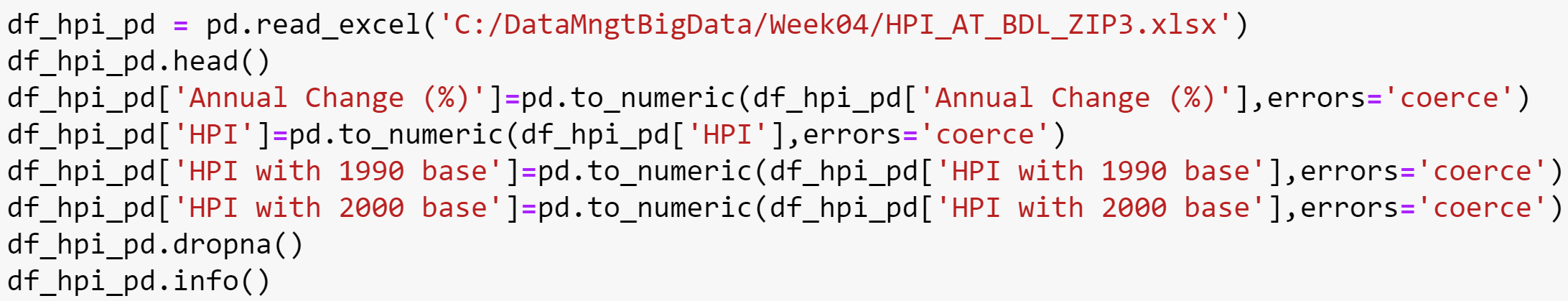


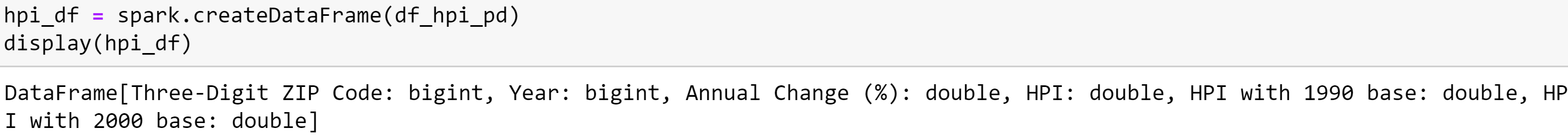
1. Distribution of PeakZHVI and ZHVI values for each city in CA state:



**PART-2: HPI\_AT\_BDL\_ZIP3 Analysis**

1. Importing dataset:The dataset is being loaded in pyspark dataframe.

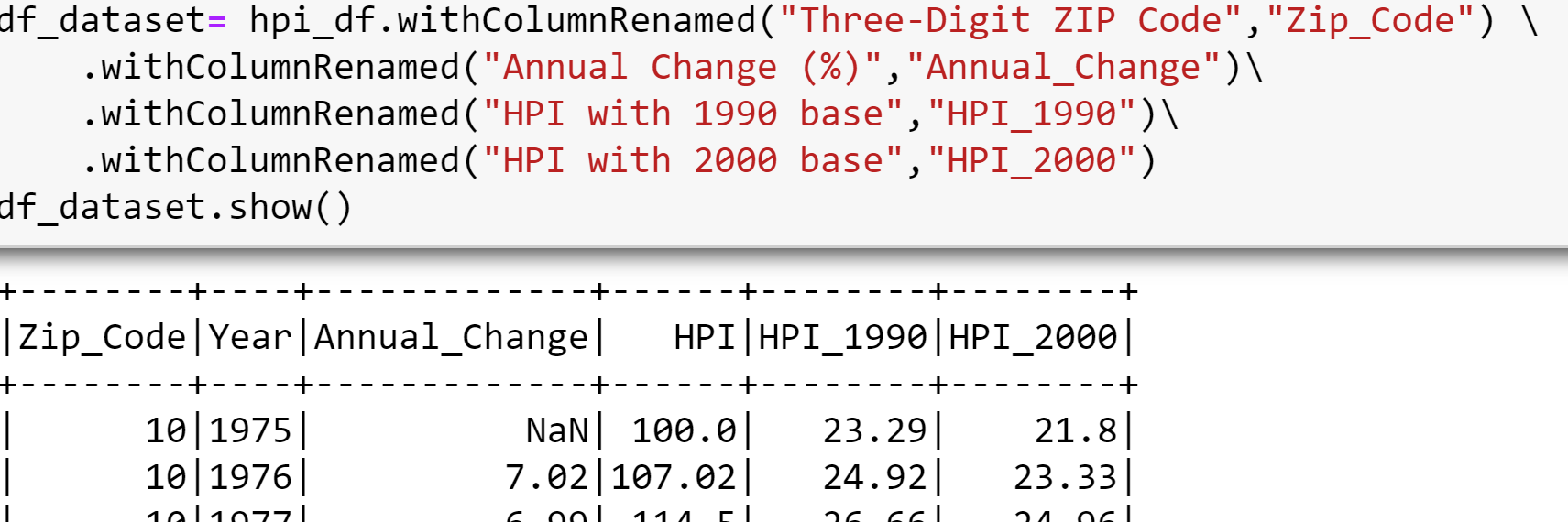




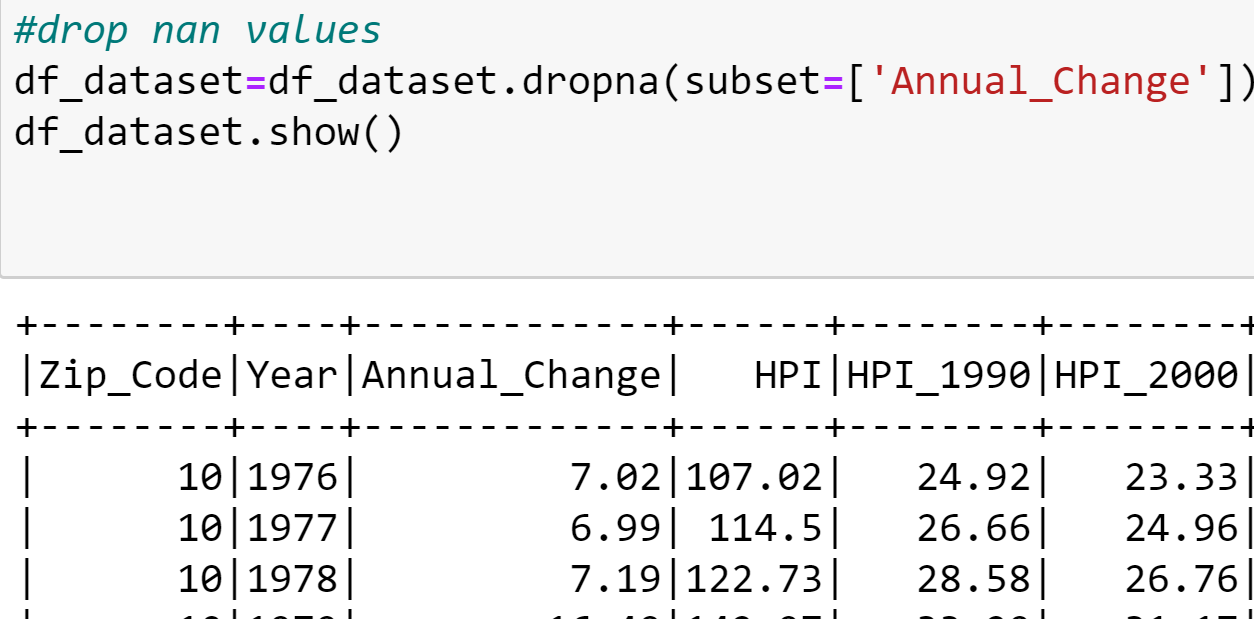
1. Temporary Table: We made the spark temporary table to run the SQL queries.



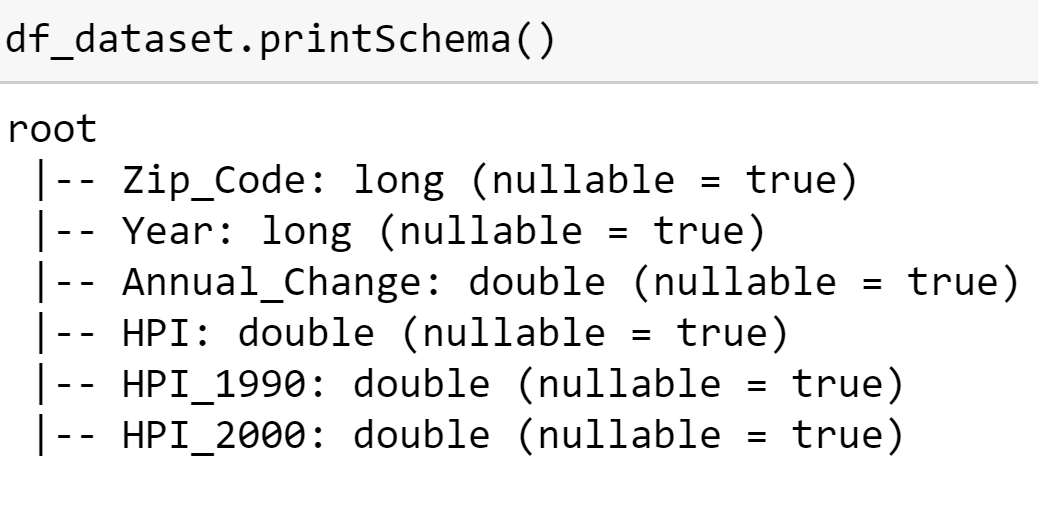
1. Renaming the columns: As we have spaces among the columns, so we renamed them:



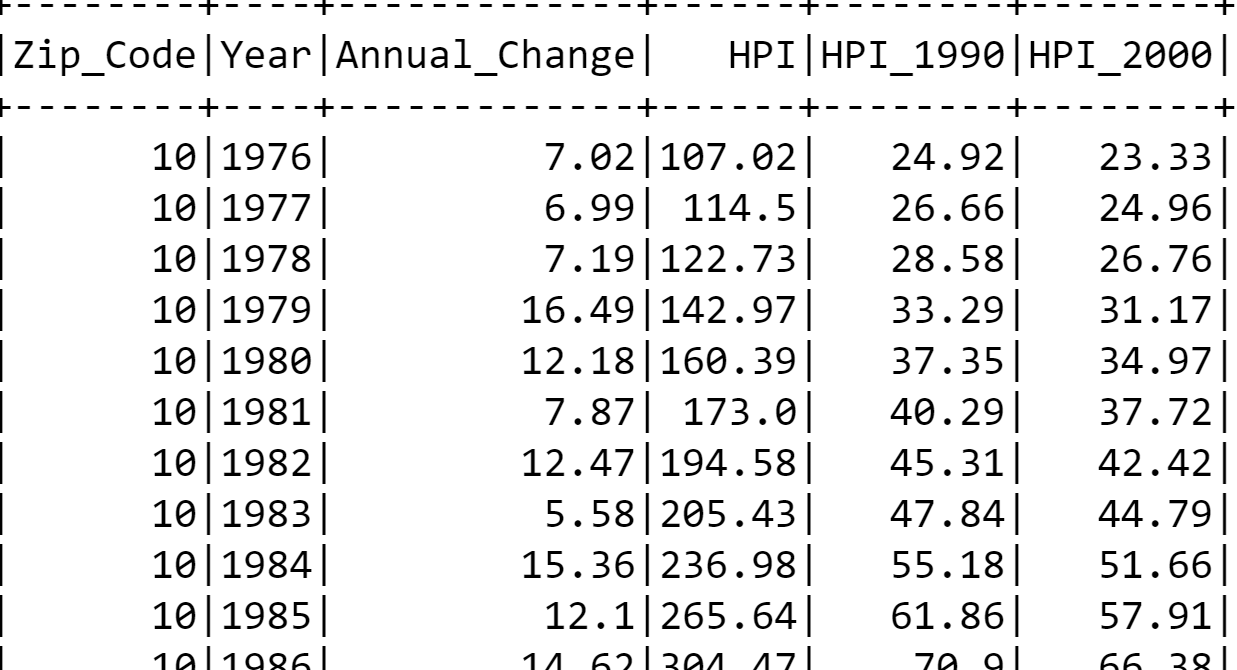
1. NaN values drop: We dropped the unnecessary NaN values present in Annual Change:



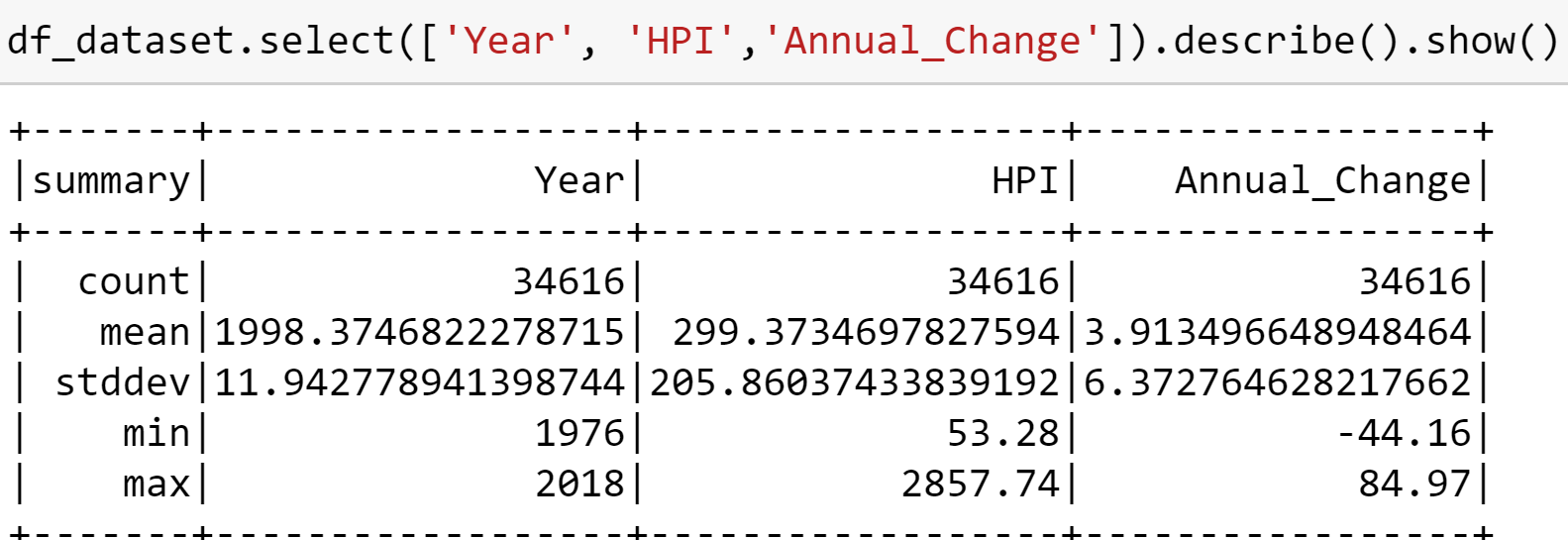
1. Schema: The printSchema function helps to view the type of coulmns in the table:



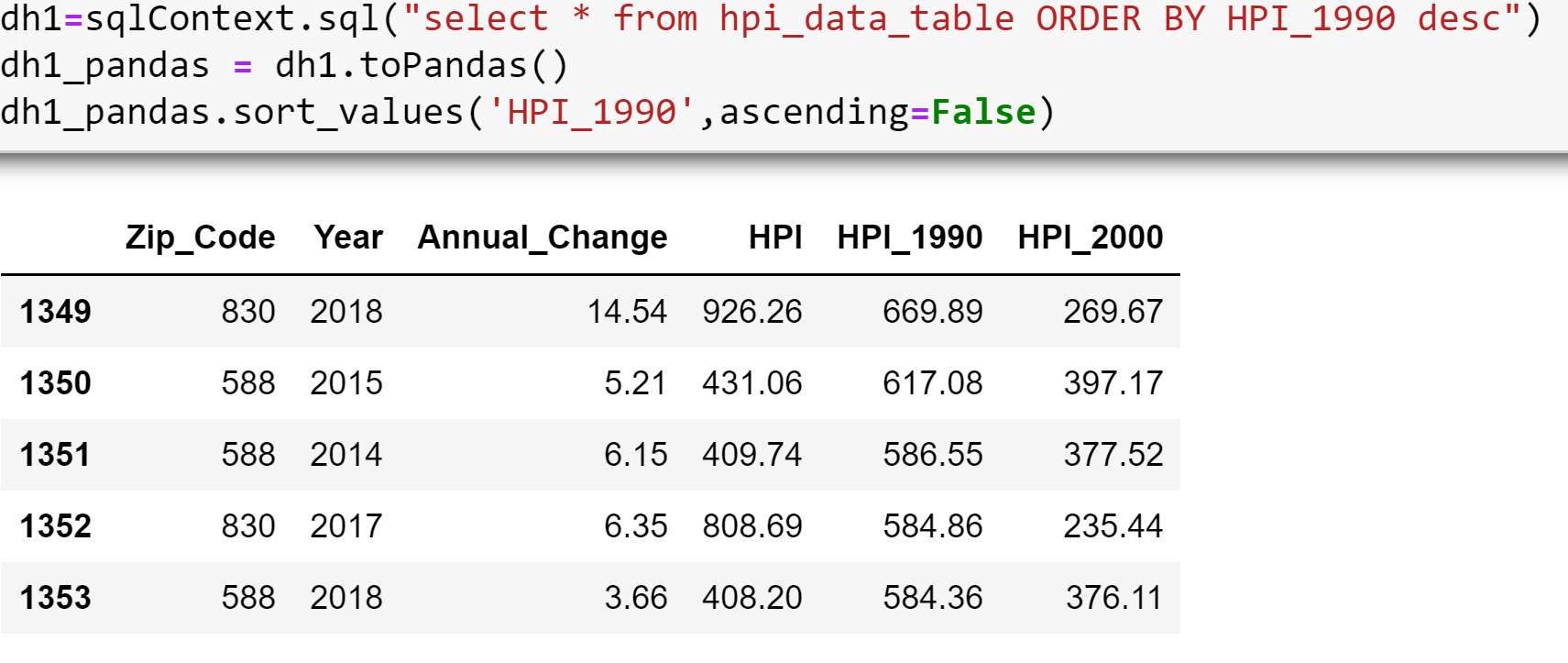
1. Displaying some rows of the table: We have used take() to display the rows of the table:



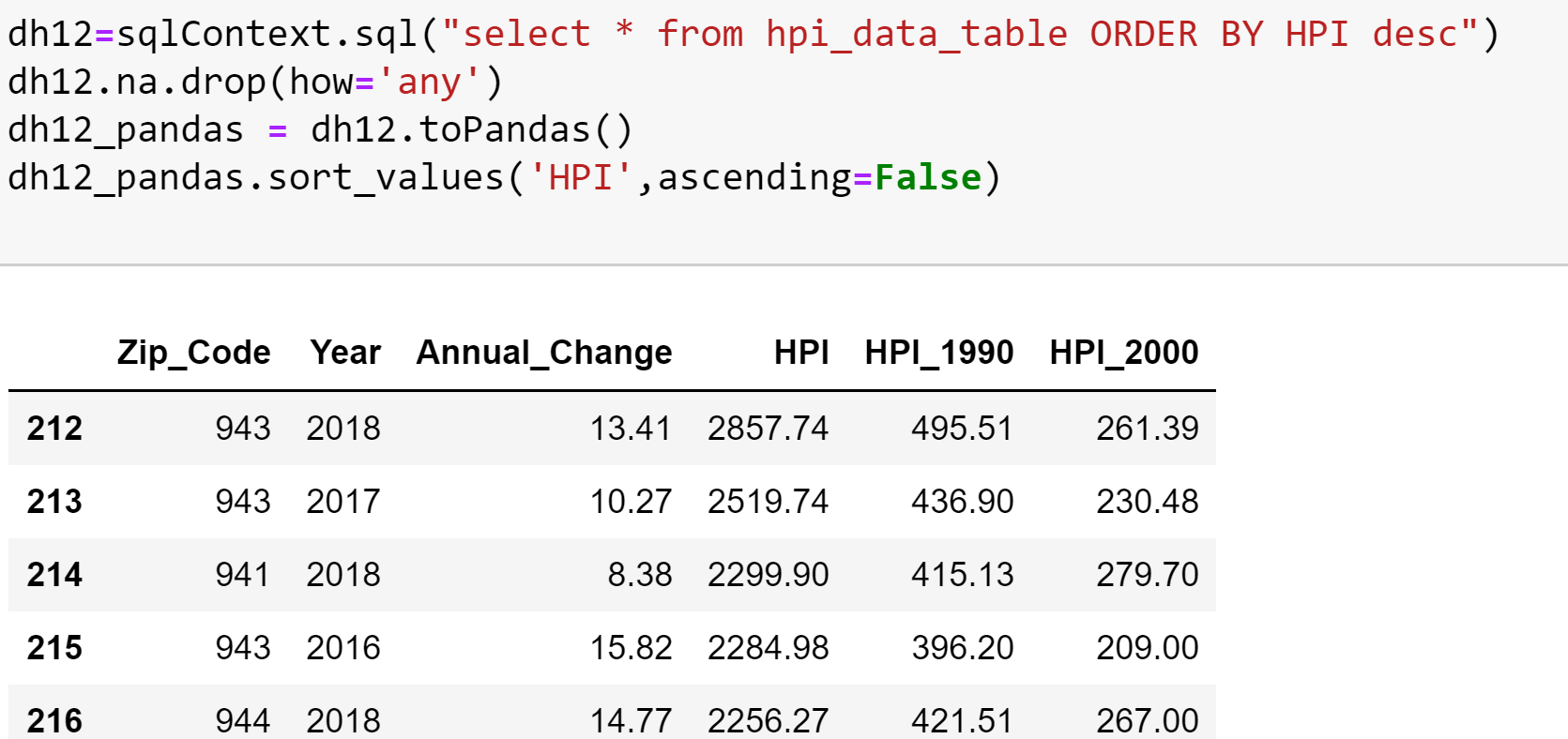
1. Information of the columns: We used describe() to show the range of the values for ZHVI columns:



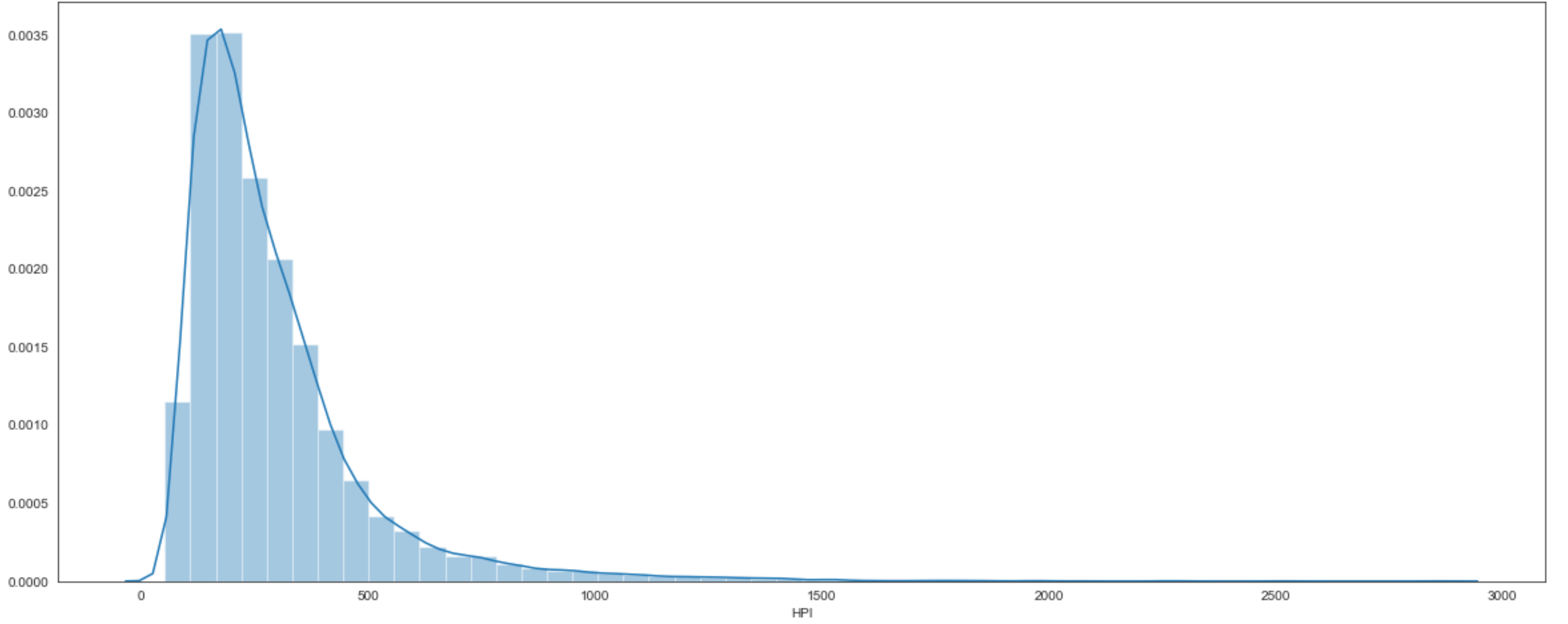
1. From the below query we found that year 2018 has the highest HPI\_1990 values:



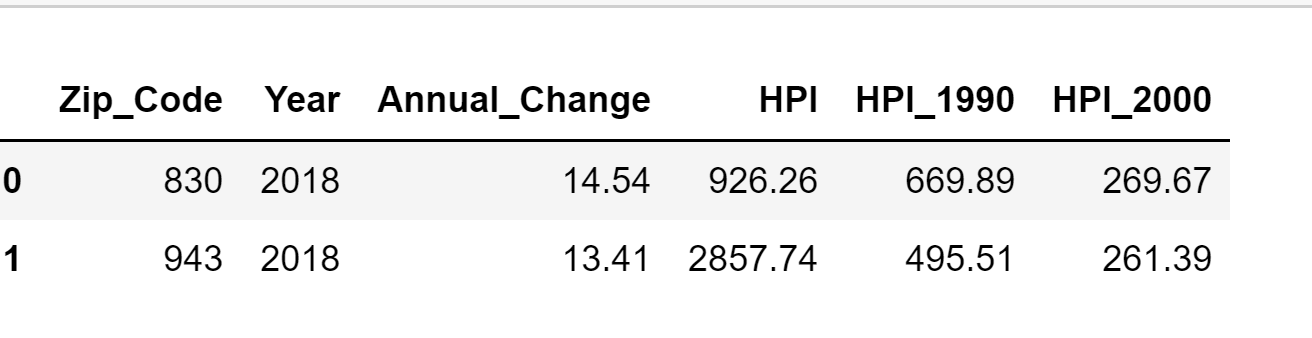
1. We get the highest HPI values.We got zip code 943 having highest value:



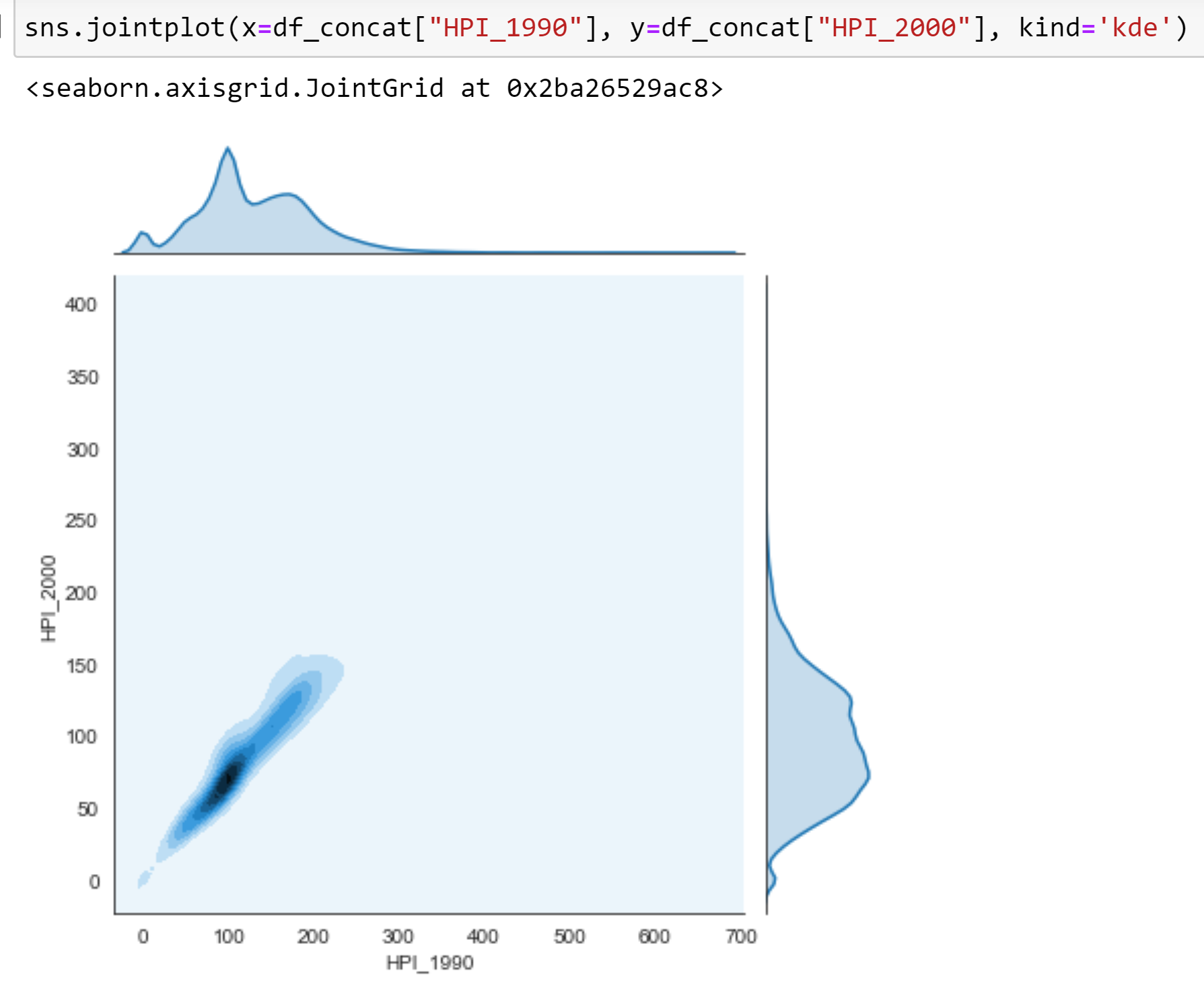
1. Distribution of the HPI values of year=2018



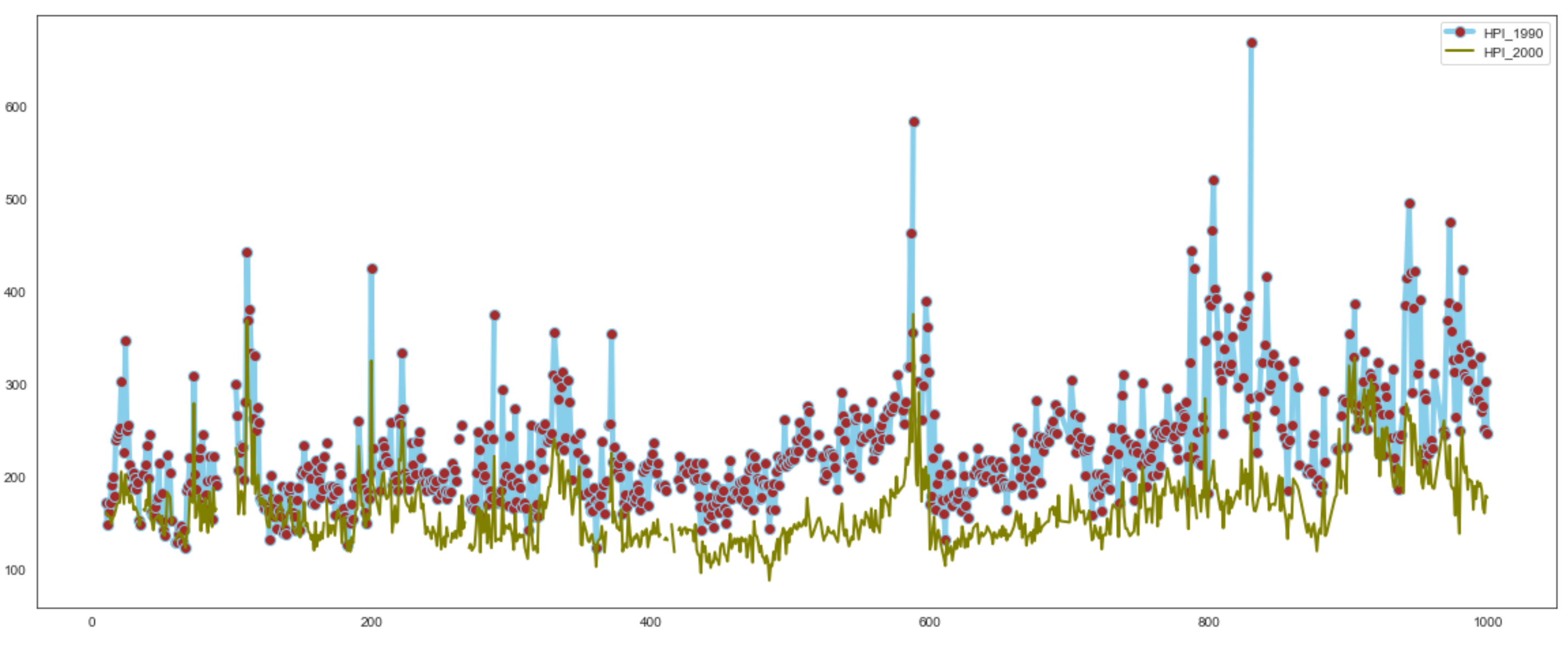
1. We have the highest HPI and HPI\_1990 values for the below areas:



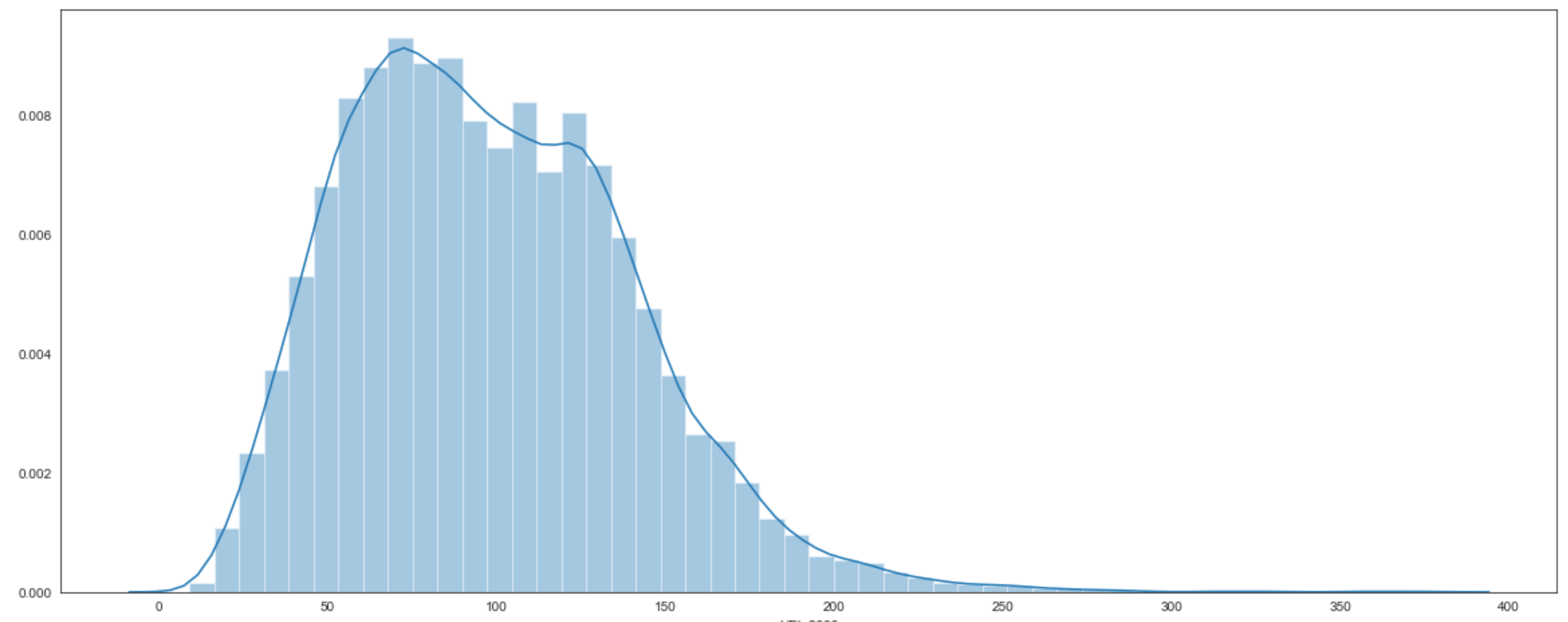
1. We digged deeper to see the distribution of the values of HPI\_1990 and HPI\_2000:



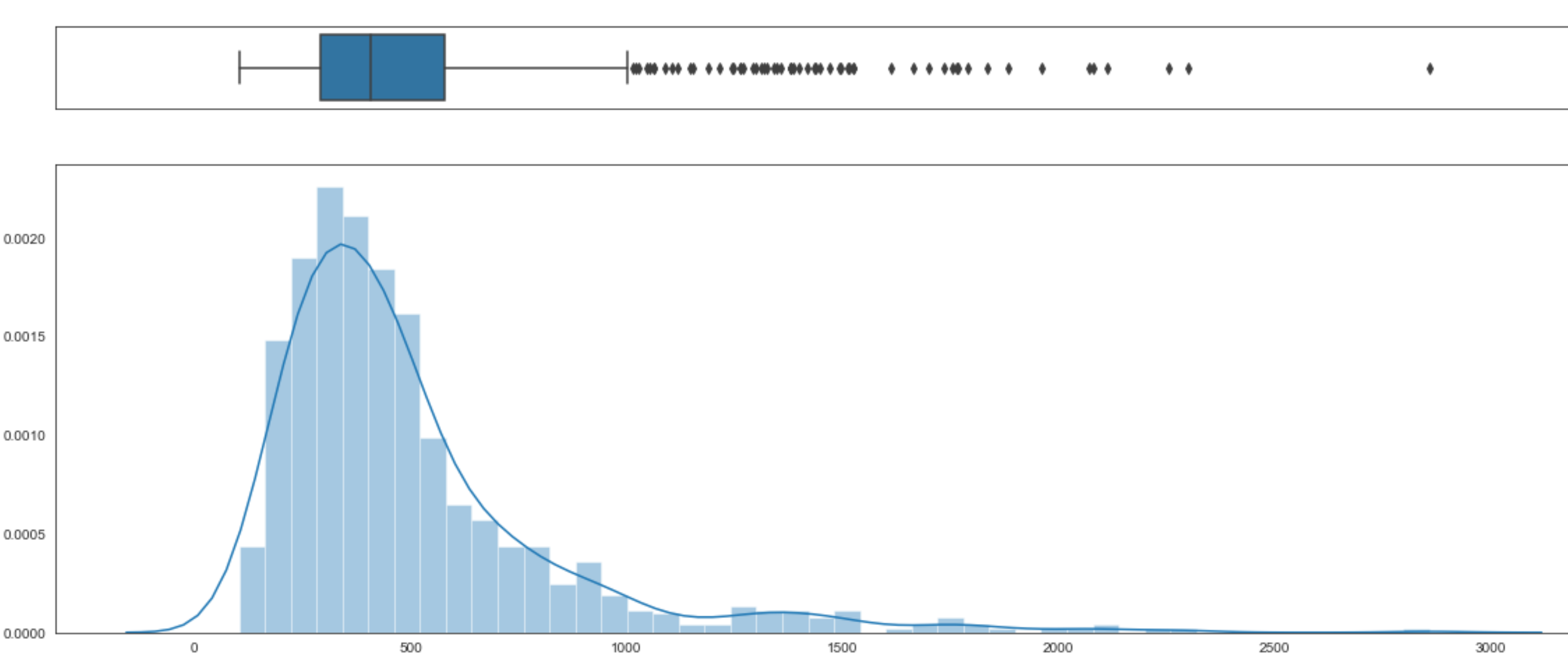
1. Distribution of HPI\_1990 and HPI\_2000 values of the year=2018:



1. Distribution of HPI\_2000 values:



1. Distribution of HPI values for year=2018 grouped by zip\_codes:



**CONCULSION**

We practiced pyspark by applying several SQL queries and getting meaningful insights. Both the datasets helped use to infer the houses with ZHVI and HPI values The process with SQL queries was way faster than analysing with normal python commands.

**REFERENCES**

1. <https://spark.apache.org/docs/2.2.0/rdd-programming-guide.html>
2. <https://data-flair.training/blogs/scala-spark-shell-commands/>