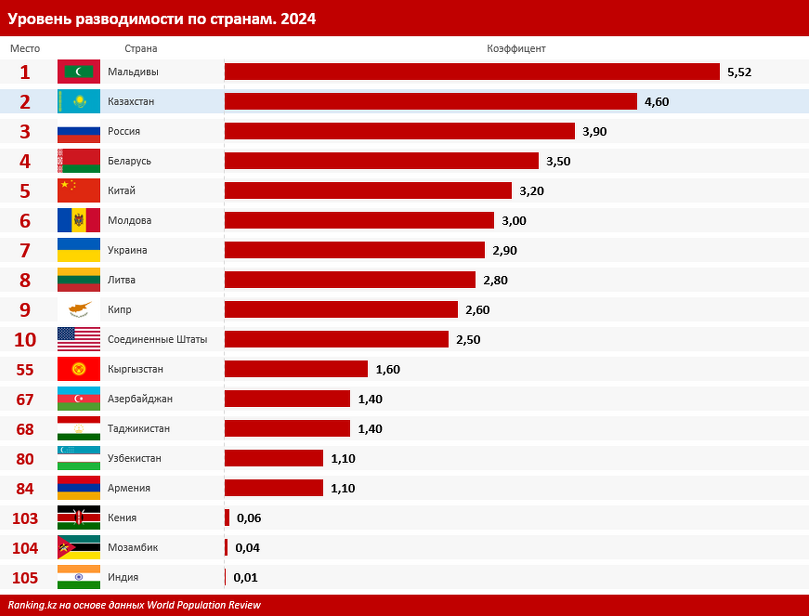
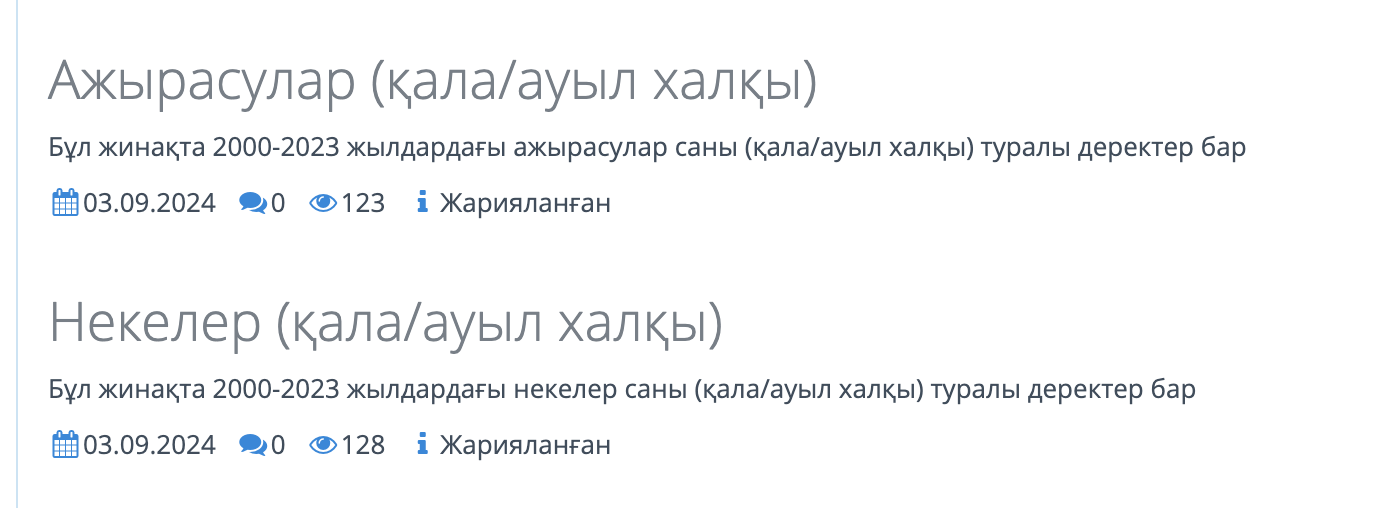
Report

Task 3: Time-Series Forecasting Objective: Build a predictive model for time-series data relevant to Kazakhstan.

**Problem:** One of the social problems in Kazakhstan is the amount of divorces.  


We see a lot of statistics on this topic and I decided to check its relevance and to predict it using machine learning algorithms.

**Dataset:**

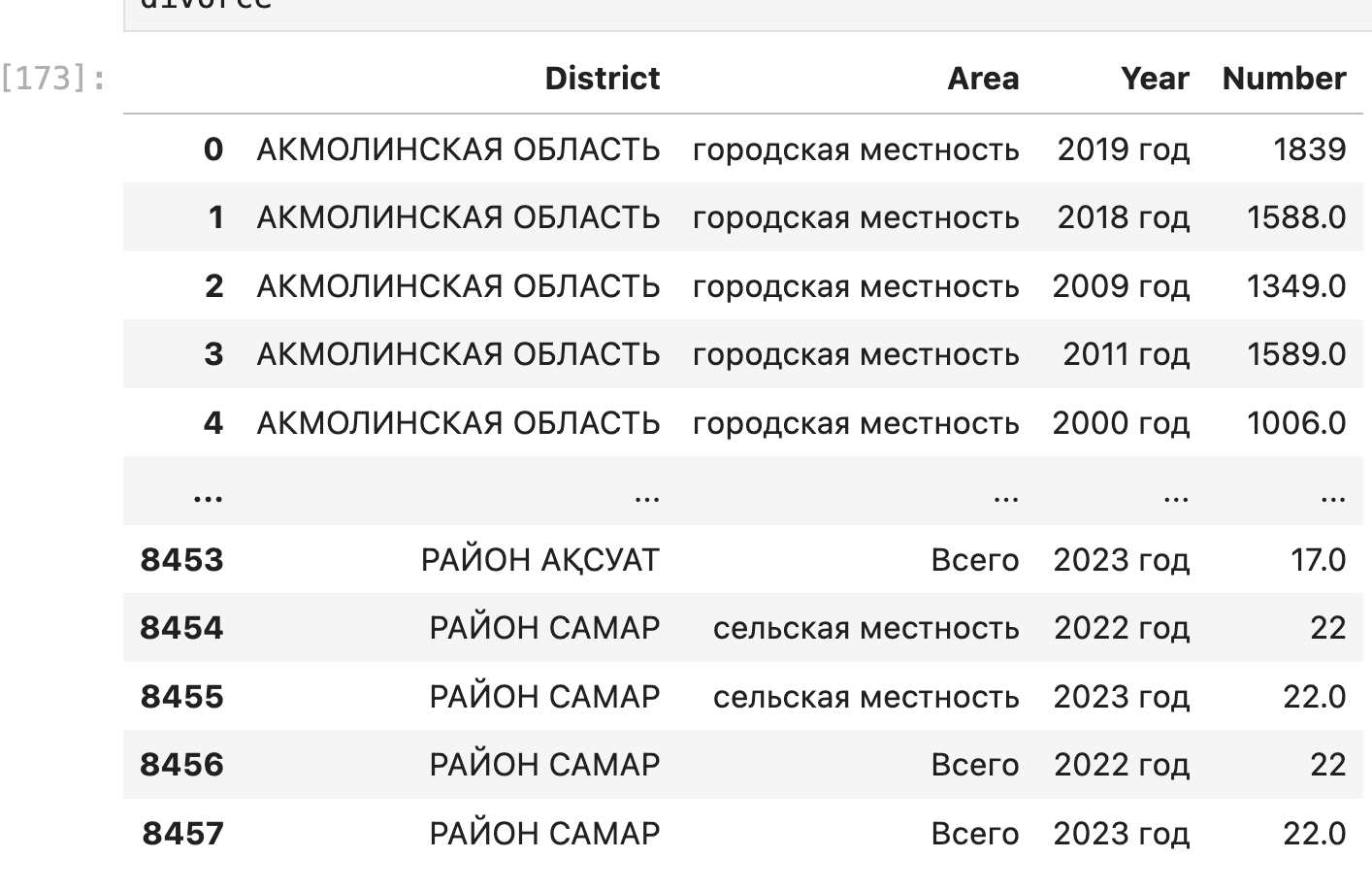
As a dataset I decided to use official data from governmental website: data.egov.kz. In the categories of population and society I found data about divorce and marriage:  


But to get these datasets I had to login to my eGOV portal, register API and only after that using that API I could download this dataset. It got in the format of .json and using external services I changed it to .csv format.

Raw format of divorce data looked something like this:  


It had 76 columns and 614 rows. It had columns ‘terms/0’ and ‘terms/1’ which are codes for region and area(rural, urban or overall). after that ‘termNames/0’ and ‘termNames/1’ which define region and area. Then ‘periods/0/name’ and ‘periods/0/date’ which are almost the same they define years. And the most important one is ‘periods/0/value’ which is the number of divorces in the specific region, area and year. After that columns continue, just changes the year and the number of divorces. This format is very inconvenient for us, that’s why we need to reshape it and make it comfortable to work with.

**Data preprocessing:**

Using different data wrangling methods like melt, pivot\_table, rename, drop and so on we made our dataset look like this:  


Now it has 8457 rows and 4 columns. Now we have a proper format of the table where each year is divided by rows not by columns. We have proper names of columns as well. Fortunately, the marriage dataset had the same format and we did all the operations on it as well. The main difference is that in number columns it has the value of marriage.

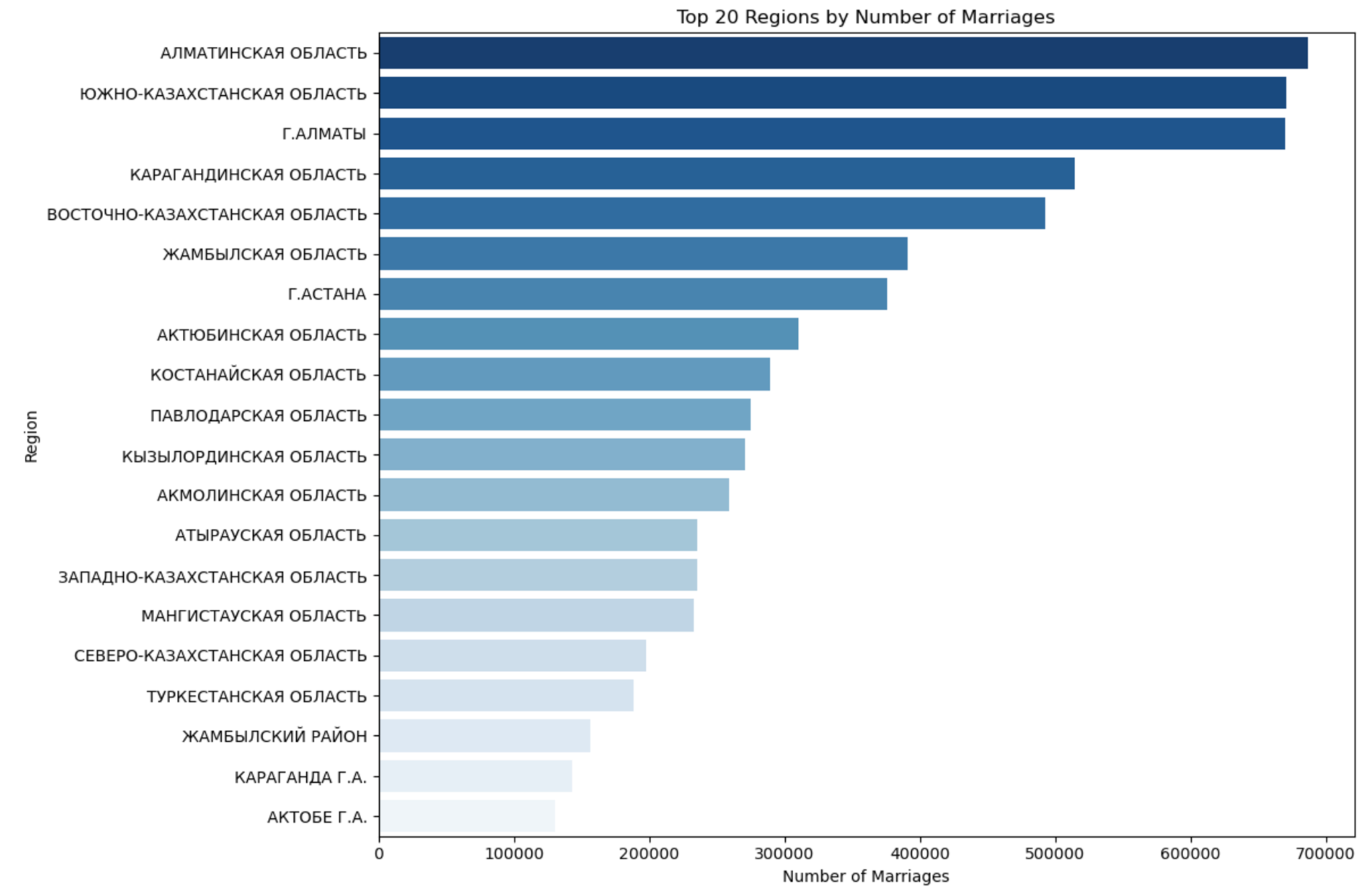
But in the end we need to merge these 2 dataframes and we get table like this:  

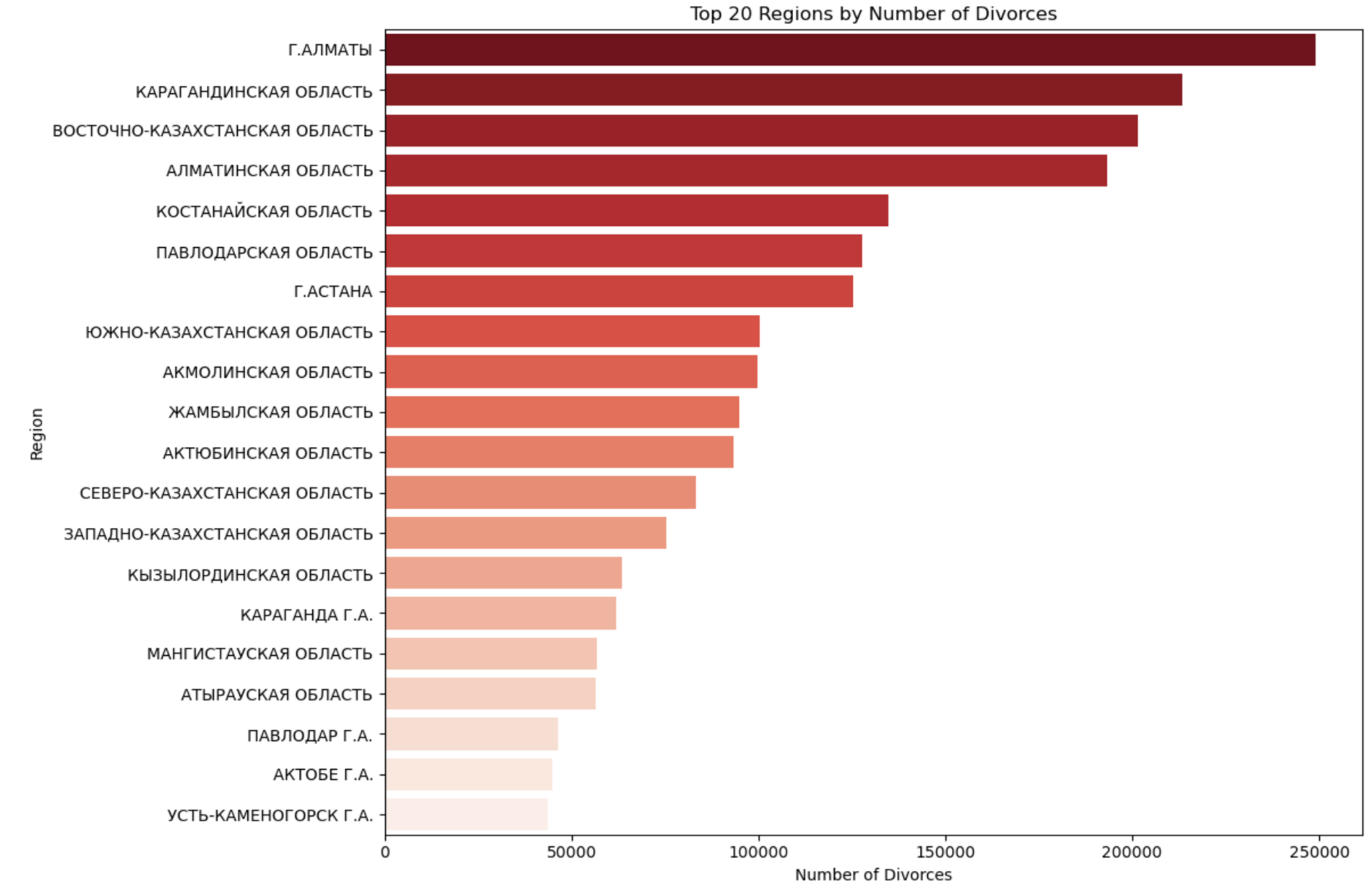

This dataset has no duplicates or null values, which makes our work easier.

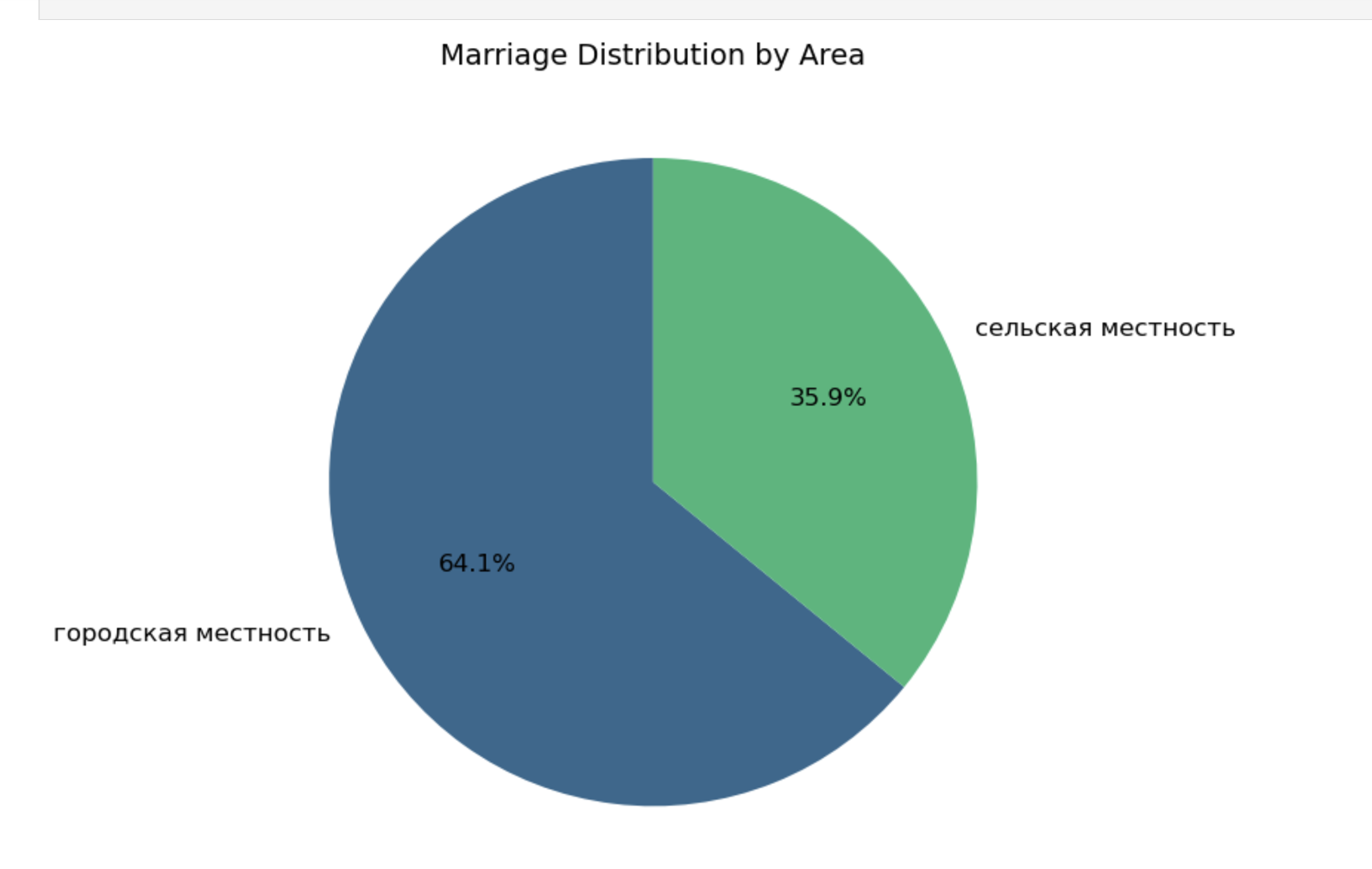
**Data visualization:**

To see basic info about our dataset we will use visualize it using matplotlib and seaborn:

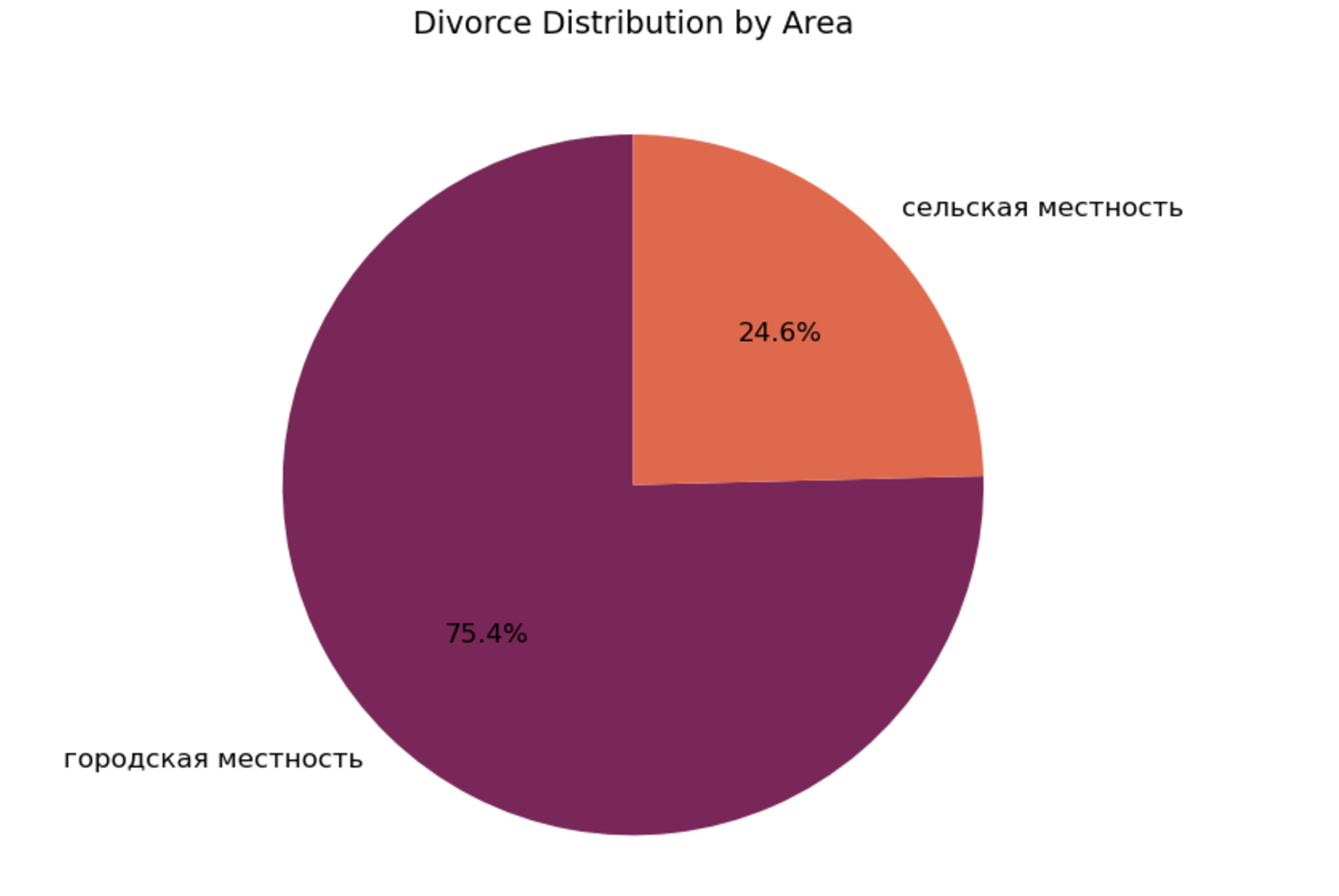
The highest one by marriages and by divorces is Almaty region. And the Karaganda region is the 2nd one by divorce. South Kazakhstan region is 2nd in marriages.

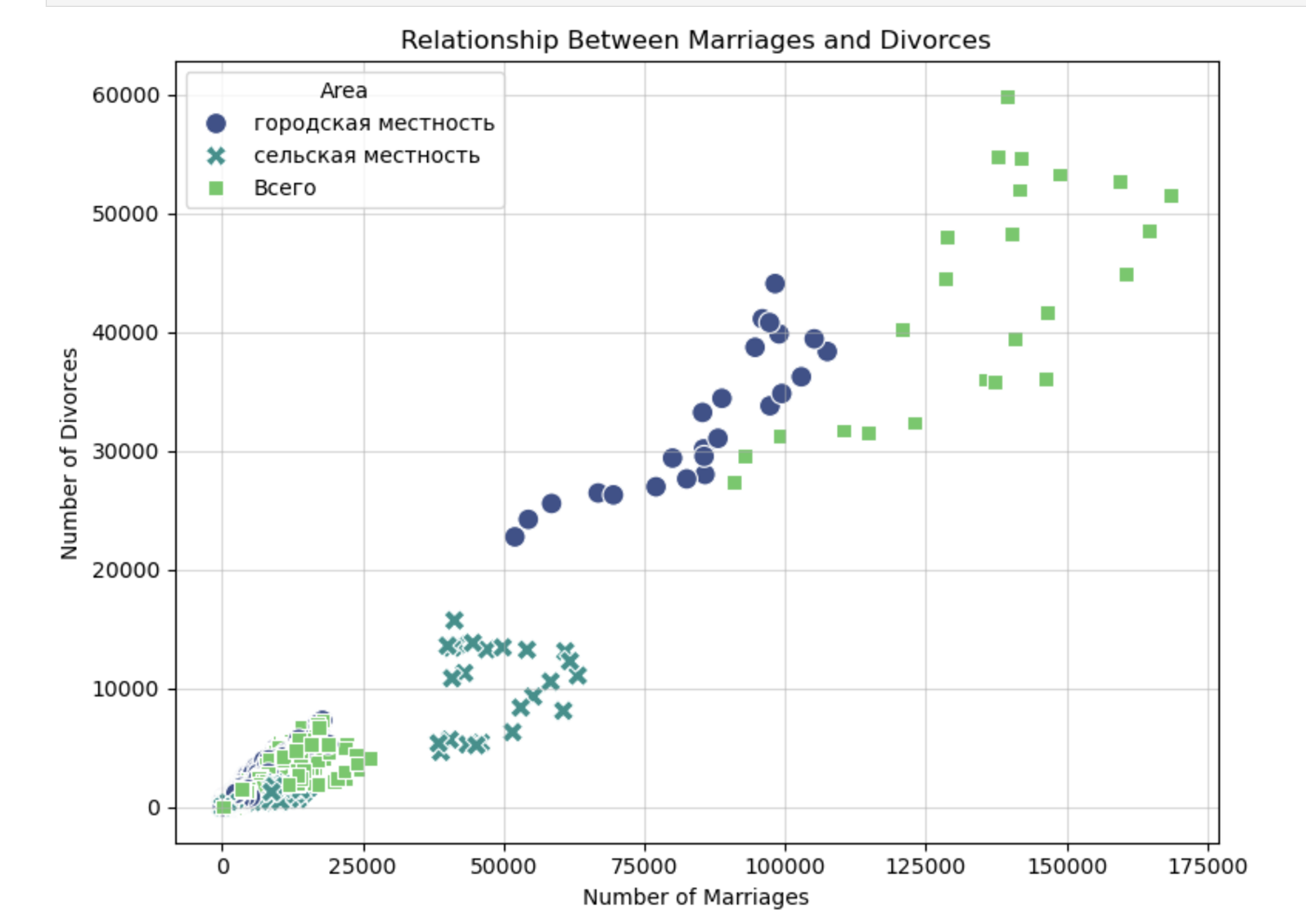






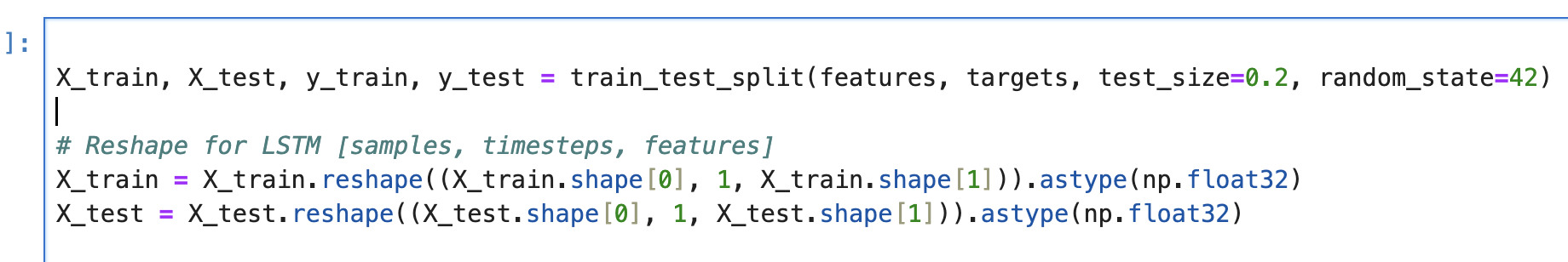
Urban divorce is 3 times higher than rural.





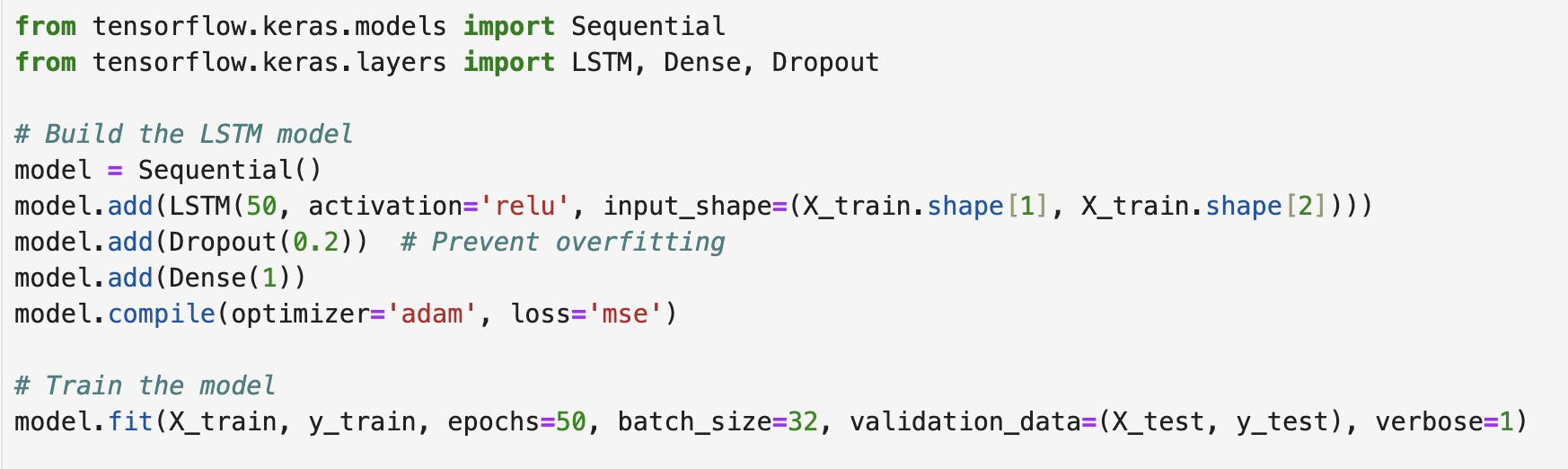
Model selection and implementation:

First of all I want to predict the number of divorce by the number of marriages in previous years. That’s why I need to create one more column. Marriages\_t-1, this column will contain data about the number of marriages in the previous years. And I will use it to predict divorce numbers. As a model I chose Long Short term memory. This model is designed to handle sequential data, making them suitable for time series prediction tasks like trends in divorces over time.

As an optimizer I am using Adaptive Moment Estimation. It balances learning efficiency and speed. But before implementing it, we split our data to train and test:  


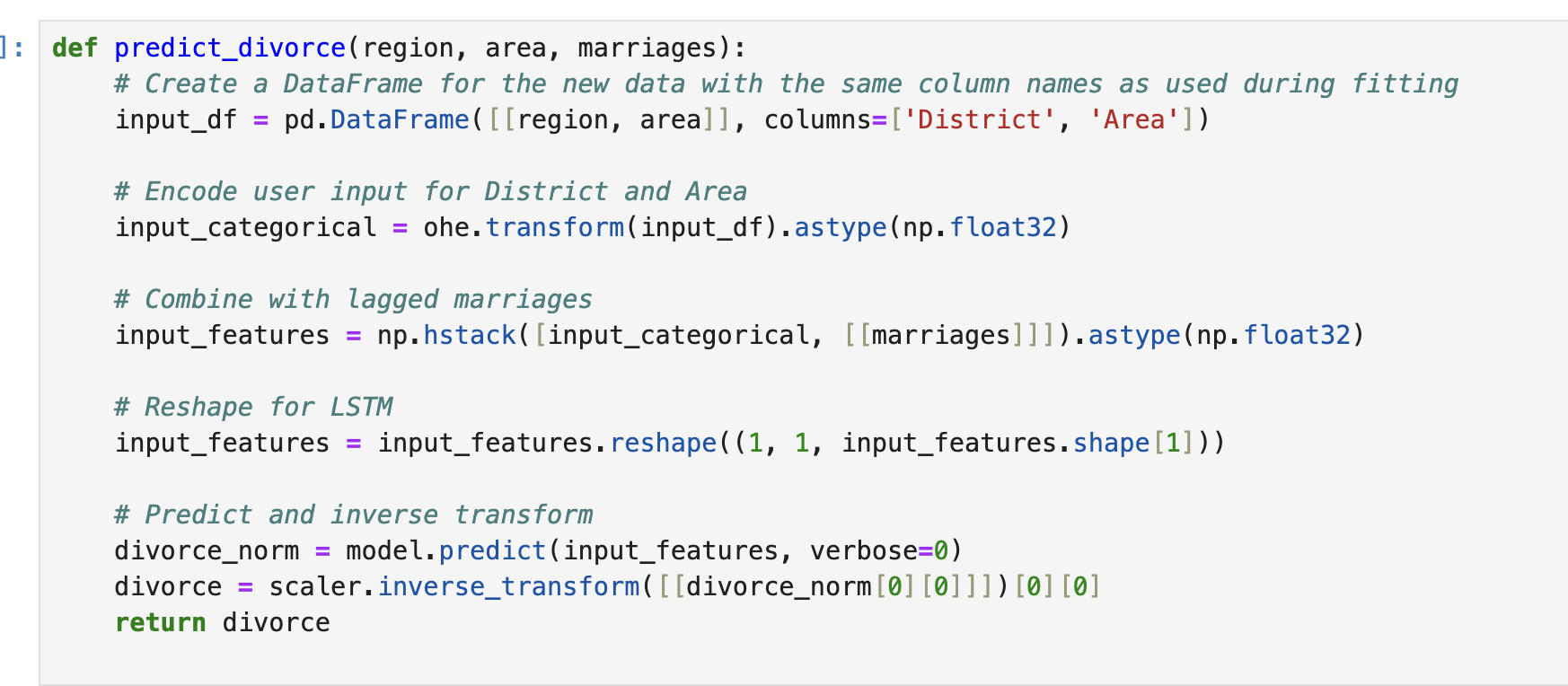
we reshape the X values because LSTMs process time series data, it must be reshaped to fit the LSTM’s expected 3D input format.

Then we import all the necessary libraries and build the model using the instruments like dropout to prevent overfitting. We use the Dense(1) because It transforms the multi-dimensional output of the LSTM into a single continuous prediction. Mean Squared Error is ideal for regression problems as it minimizes the squared differences between predicted and actual values.

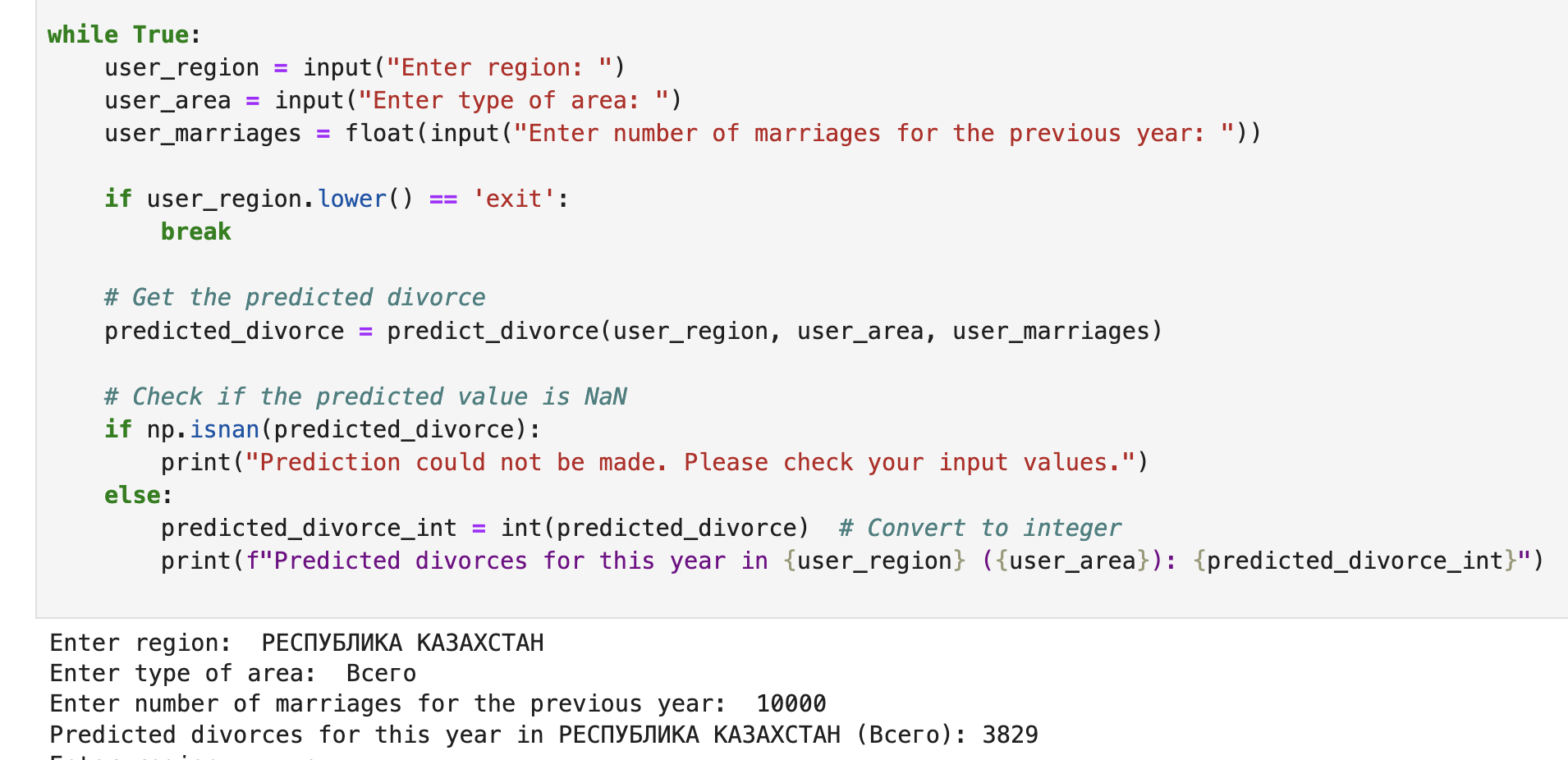


And using the .fit() function we finally train our model on the given parameters.

Now we create predict\_divorce() function so we can **make predictions** about divorces based on user-provided inputs: region, area, and the number of marriages.



And now function that can show us results of our work:



Using the trained model and some operations we can now predict how many divorces we might face in the next year.