**Voter Support Classification on Government Fiscal Responsibility**

1. **Introduction**

In this project, a hundred anonymized features are given to predict whether a voter is likely to support higher government spending versus reducing the national debt when given a choice. The provided data contain survey responses from 20,000 randomly selected individuals across the country. There are two responses: Spend to Improve Economy (represented by 1) or Reduce National Debt and Deficit (represented by 0). The goal of this project is to predict the responses for 29,231 voters.

1. **Data exploration and feature engineering**

This is a binary classification problem. There are 20,000 samples in the training dataset. The training dataset is a subset of the 29,231 records, which will be predicted at the end. In this case, the test dataset has instead 9,231 records. The available features include 24 categorical features and 124 numerical ones. The distributions of the both kinds of features are shown along with the python code in Jupyter notebook.

It can be observed that a large amount of categorical features contain more than 10 values. Label encoding and One hot encoding largely increase the dimension of the problem. Frequency encoding is also performed for high cardinality features (State, f1, f115 and f121).

The numerical features are highly skewed. First, the skewness of each numerical feature is computed. Then a box-cox transformation is applied on features with skewness greater than 0.25.

The labels have a ratio 0.7:0.3 for classes 0:1. It is not a highly imbalanced problem. There is no need to either resample the data or apply weight in the models.

There also exist quite a lot of missing values. Impute of missing values usually requires domain knowledge. Since the features are all anonymous and encoded, missing values are not treated here.

It is also worth noticing that the patterns of some features are different between training and testing dataset. This may potentially affect the accuracy of the prediction.

1. **XGBoost model**

An XGBoost model is built for this problem. XGBoost is a powerful and popular model due to its good performance. It is widely used, especially in the Kaggle competition. The model is constructed to improve the predicting error from existing ensemble, as shown in Figure 1. It also has a special way to treat missing values and also requires less pre-processing on data. It is used here because of these reasons.

To build a better model, a 5 fold cross validation is used to tune the parameters. Parameters (max\_depth, min\_child\_weight, colsample\_bytree, and gamma) are first tuned manually, with a large learning rate. Then Bayesian optimization is used for automated parameter tuning. With the optimized parameters, a model is trained using a smaller learning rate. A prediction is made using this model.

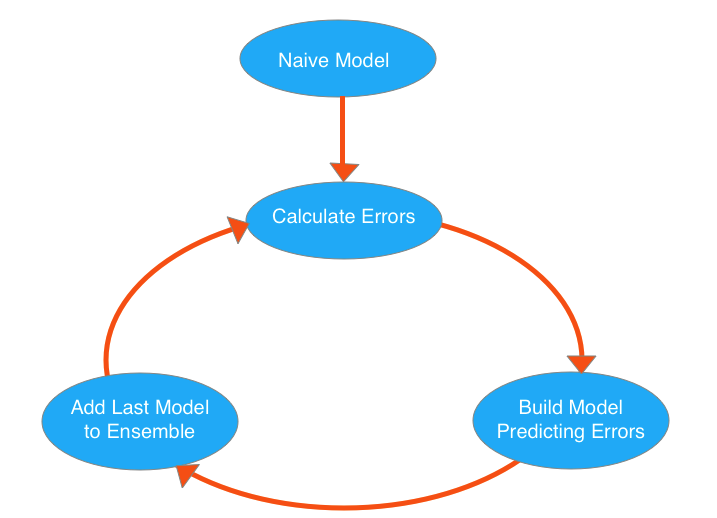


Figure 1. XGBoost model

The default evaluation metric (‘error’) for binary classification is used in this process. Part of the feature importance is shown in Figure 2.

The values of evaluation metric have small difference for different parameters. To explore the possibility of improving the model, feature combination is tried. For categorical features, combinations of any 2 are created. This process increases the dimension to over 4000. However, there is no obvious benefit observed. Considering the increased computational cost, the combinations of categorical features are not used in the final model.

For binary classification, the output prediction of XGBoost is probability confidence scores in [0,1]. To get the classification result, the training dataset is splited to two parts with ration 0.8:0.2 randomly. The first part is used to find the probability threshold for the classification. The threshold obtained is 0.32. Then the model performance is evaluated on the second part. The performance measures are discussed in the next session.

At the same time, the classifications for all the 29,231 records can also obtained by applying this threshold.

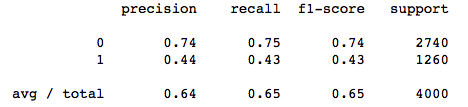
1. **Model performance measures**

Three measures are checked to evaluate the mode. First, the accuracy is 0.6475. Accuracy is ratio of the correct classifications vs. all populations. It is natural to get an impression on the performance of a model from accuracy. But it doesn’t tell the whole story. So confusion matrix and classification report are also output here.

The confusion matrix is



And the classification report shows



These measurements are reasonable for a binary classification model.

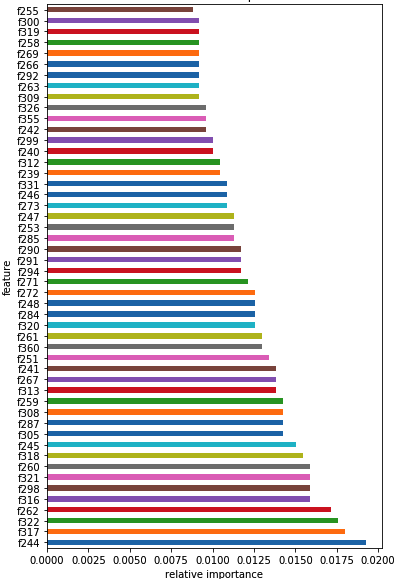


Figure 2. XGBoost feature importance

1. **Future improvements**

There are spaces to improve the classification, based on the measurements in the previous session. It is observed that the probability confidence scores have a small range. There are also no features with dominant importance.

To improve the classification, further understanding of the input, and feature engineering are the most important 2 directions. It is also worth to try different models, especially the modern algorithms. Building an ensemble model can also improve the result.

1. **Output visualization**

The mean values of classification and the 5 features with largest importance (f9, f84, f89, f29 and f83) are overlaid to states except Alaska. There classification results visually have common patterns with features to some extent.

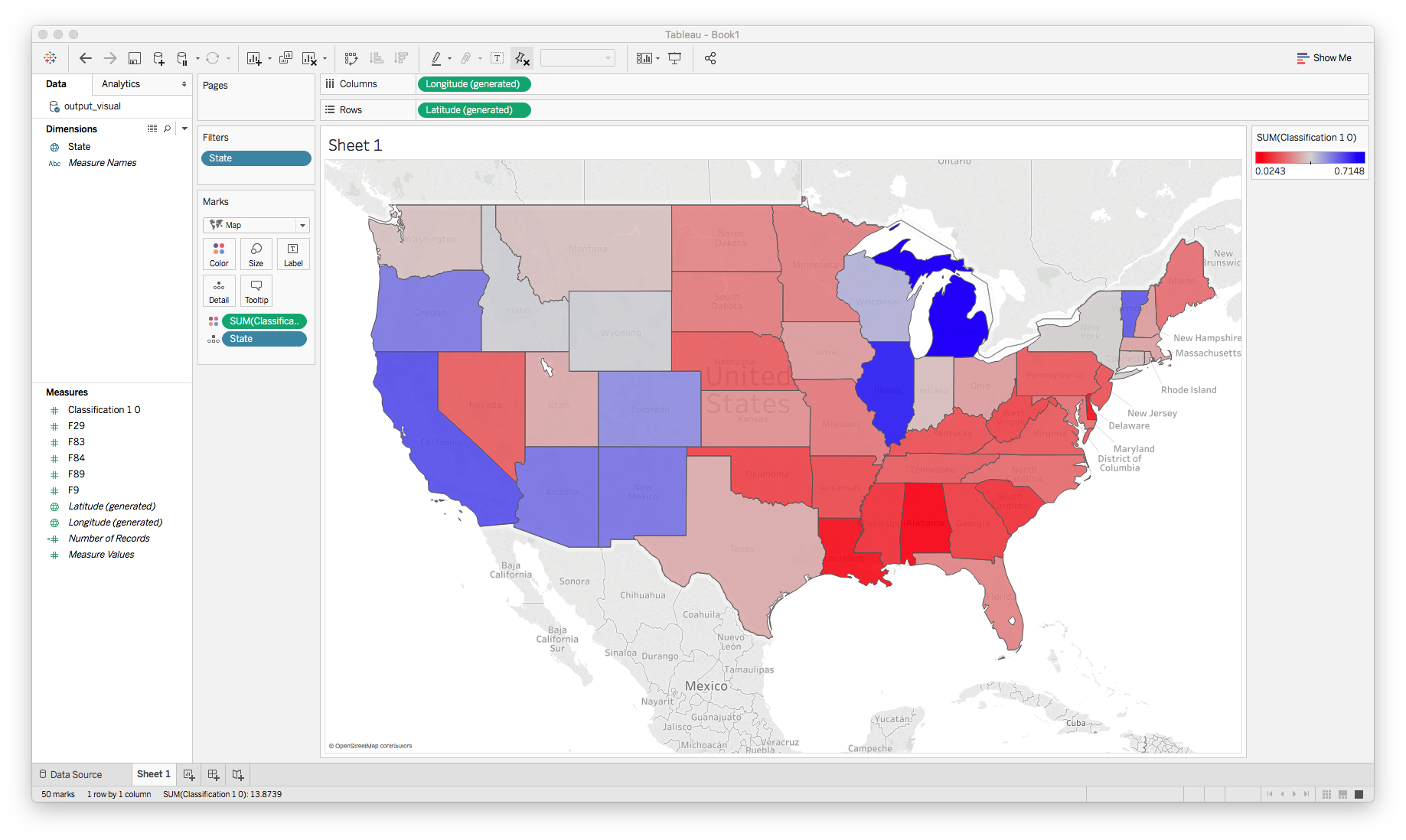


Figure 3. Mean values of predicted classifications

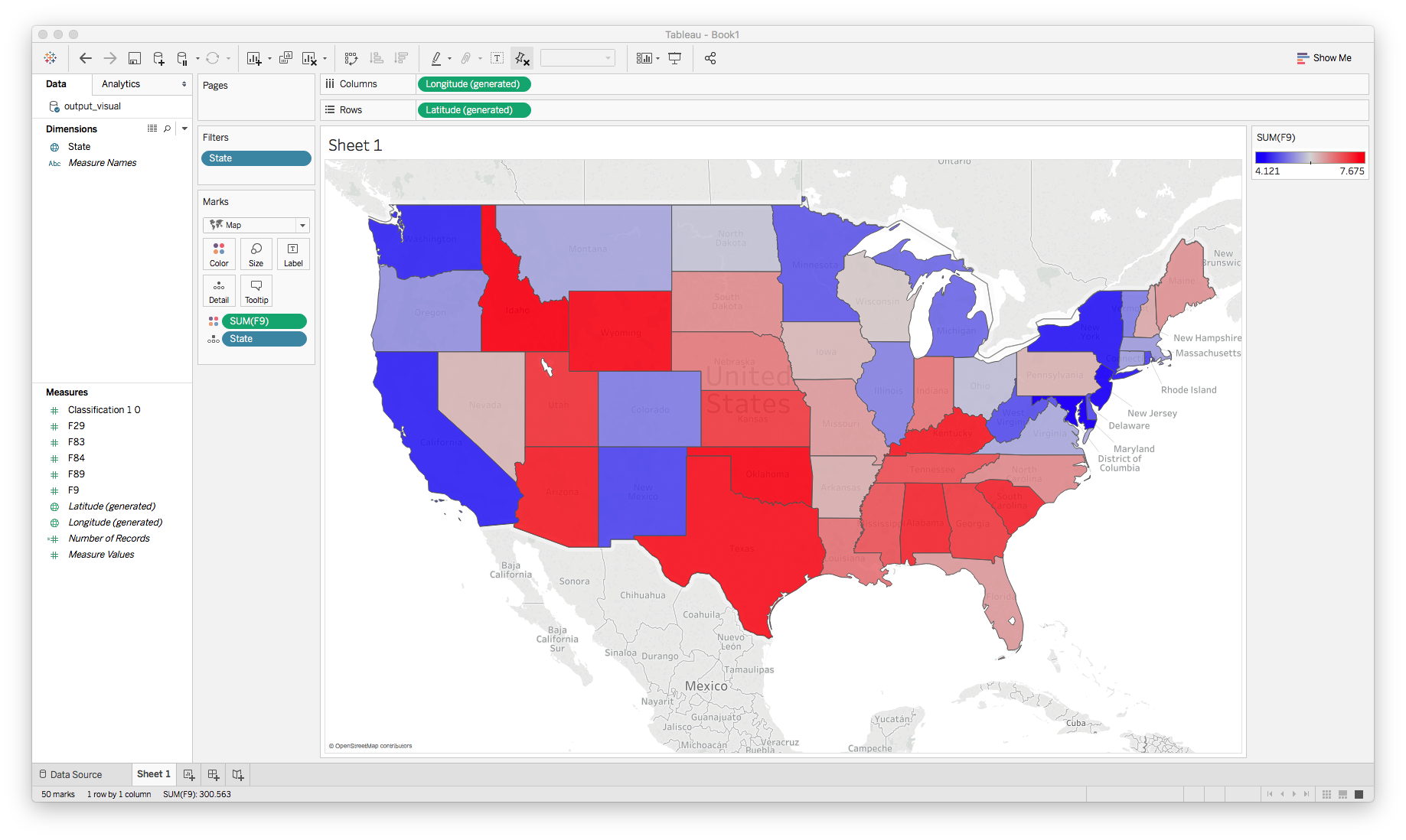


Figure 4. Mean values of feature f9

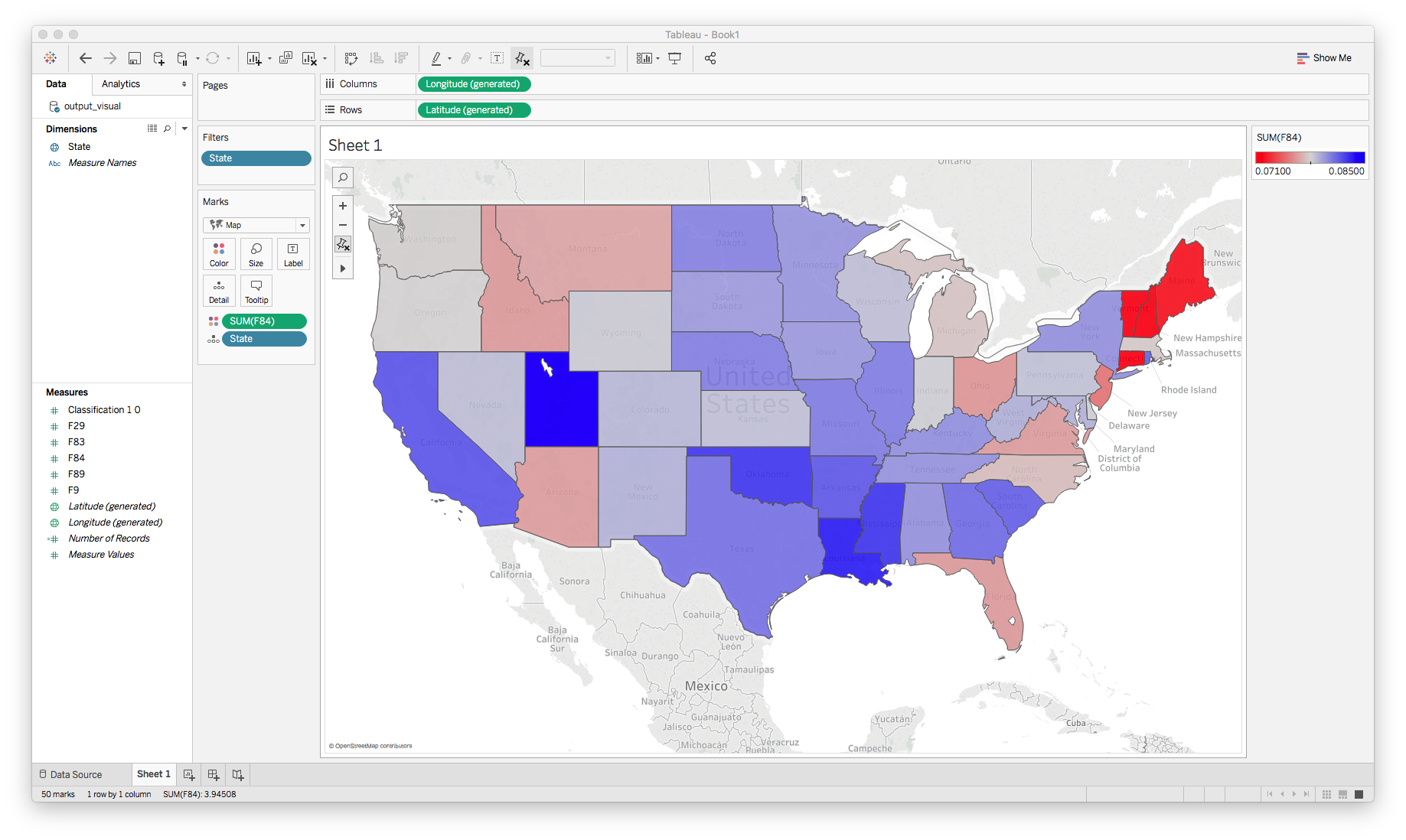


Figure 5. Mean values of feature f84

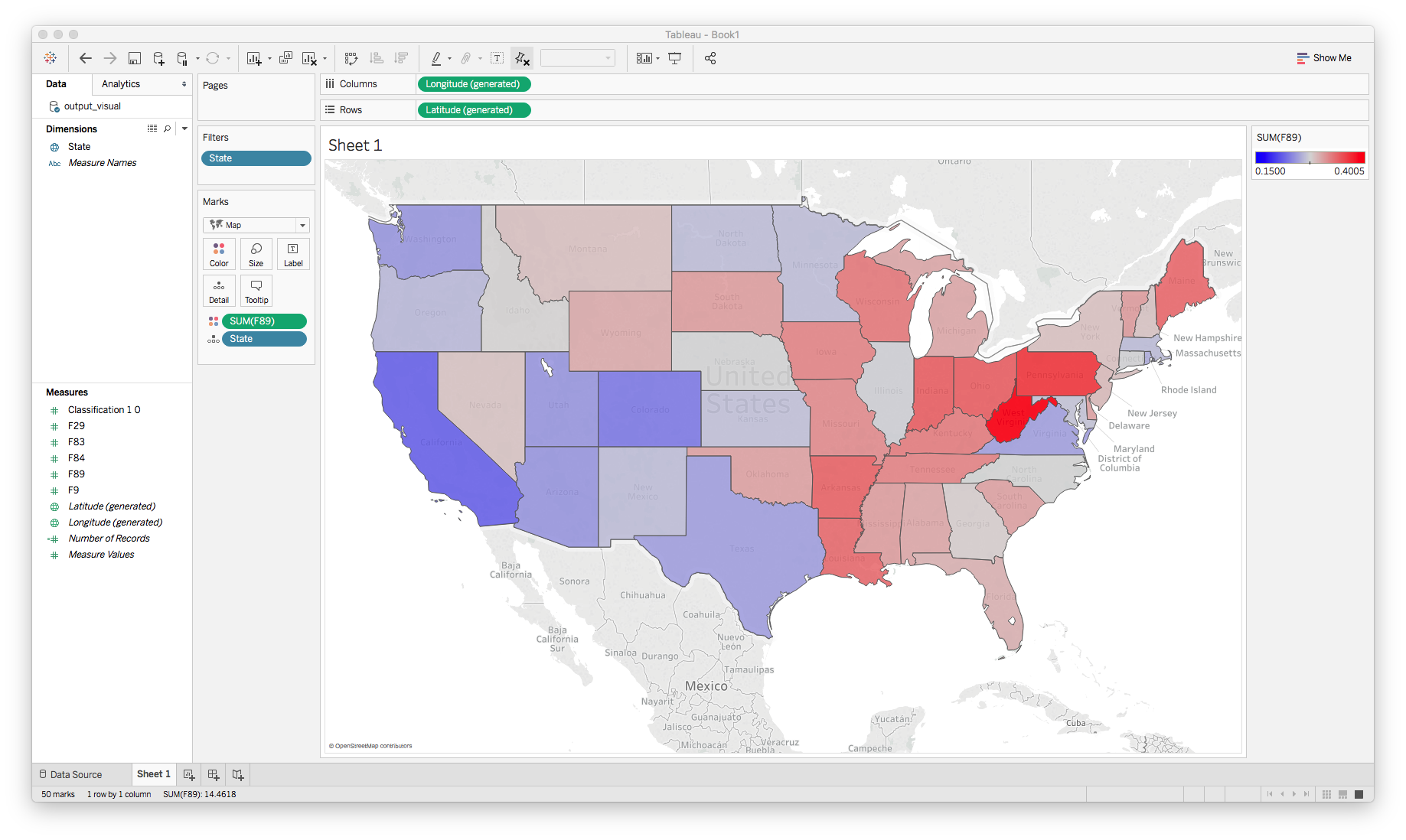


Figure 6. Mean values of feature f89

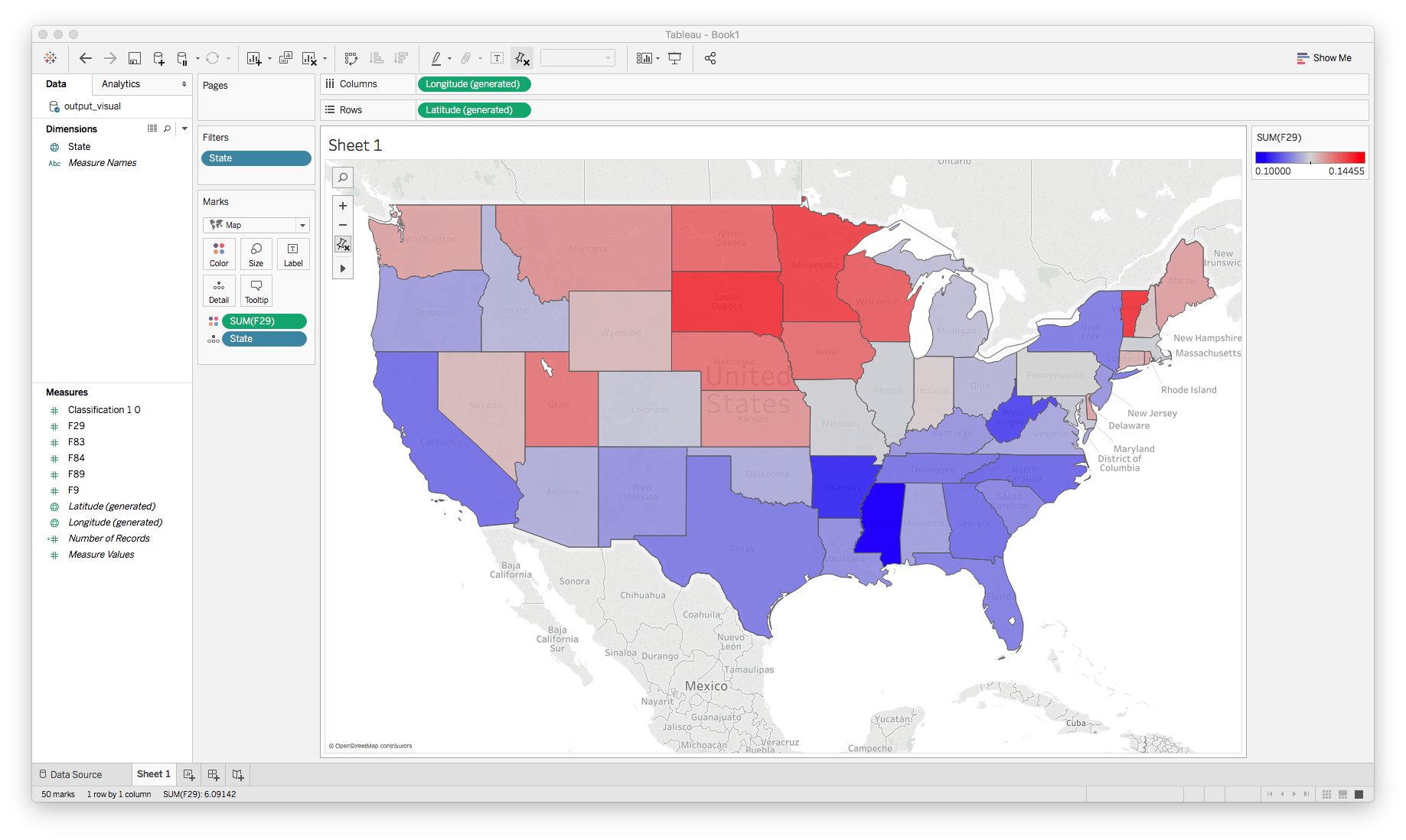


Figure 7. Mean values of feature f29

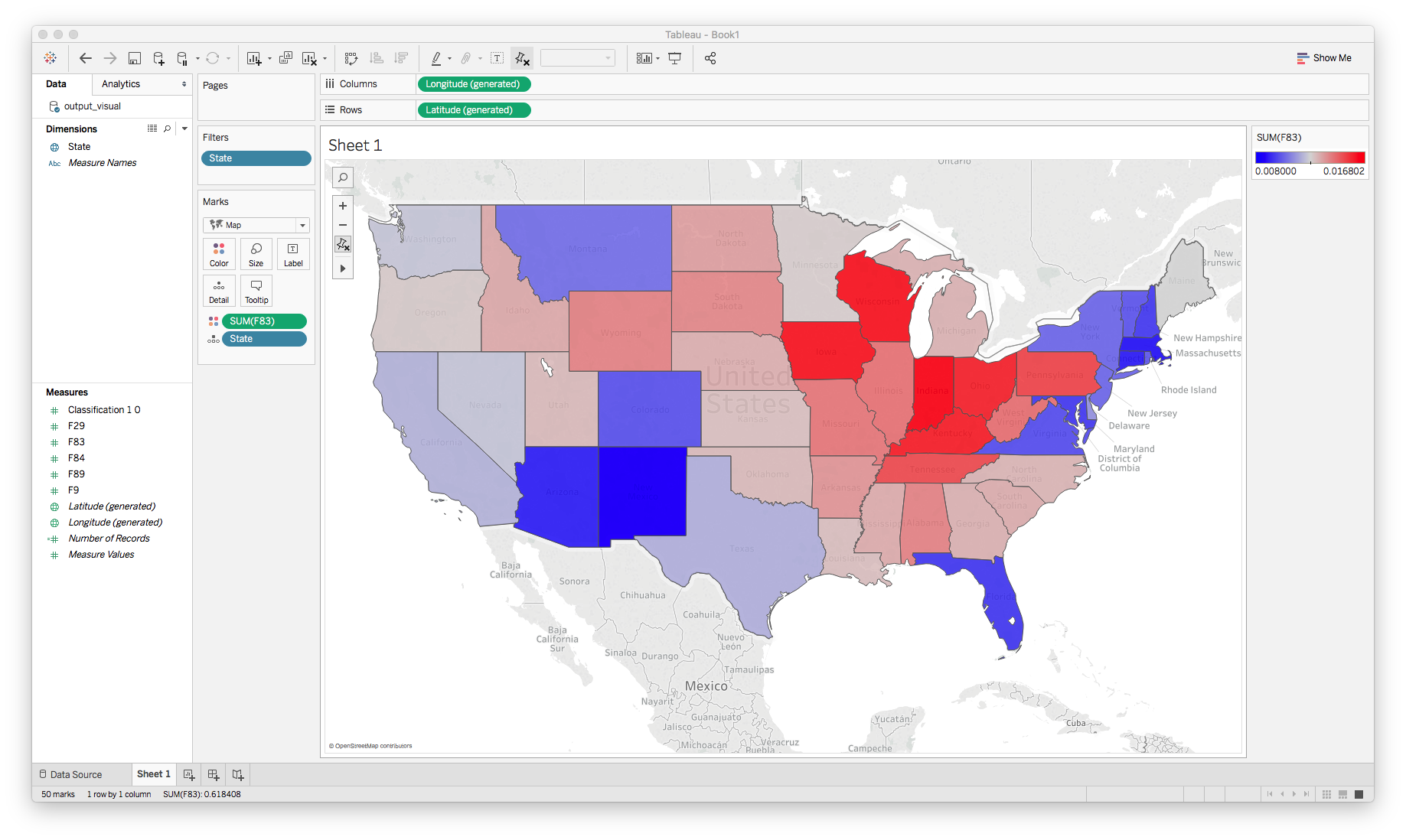


Figure 8. Mean values of feature f83