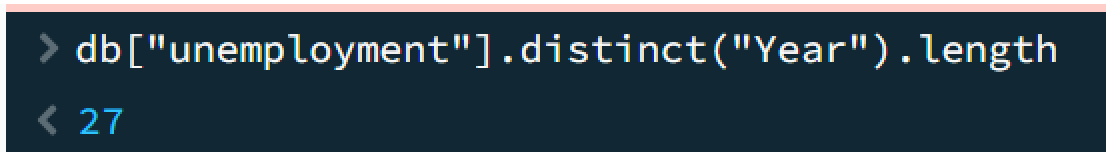
DB Assignment 5

Sean Gor

11/22/2024

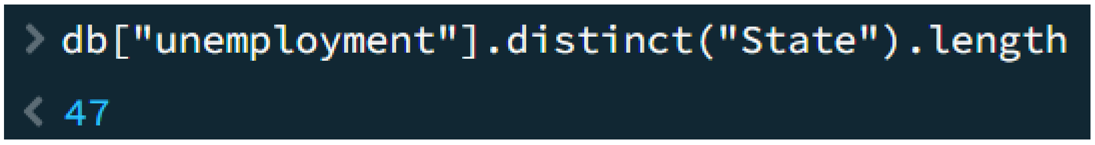
Query 1: Over how many years was the unemployment data collected?



```
> db["unemployment"].distinct("Year").length
< 27
```

This query uses the distinct function to filter all unique years. It then uses the length function to obtain how many unique years there are.

Query 2: How many states were reported on in this dataset?

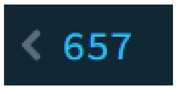


```
> db["unemployment"].distinct("State").length
< 47
```

This query uses the distinct function once again to filter the unique state names. The length function is used to obtain the number of those filtered state names.

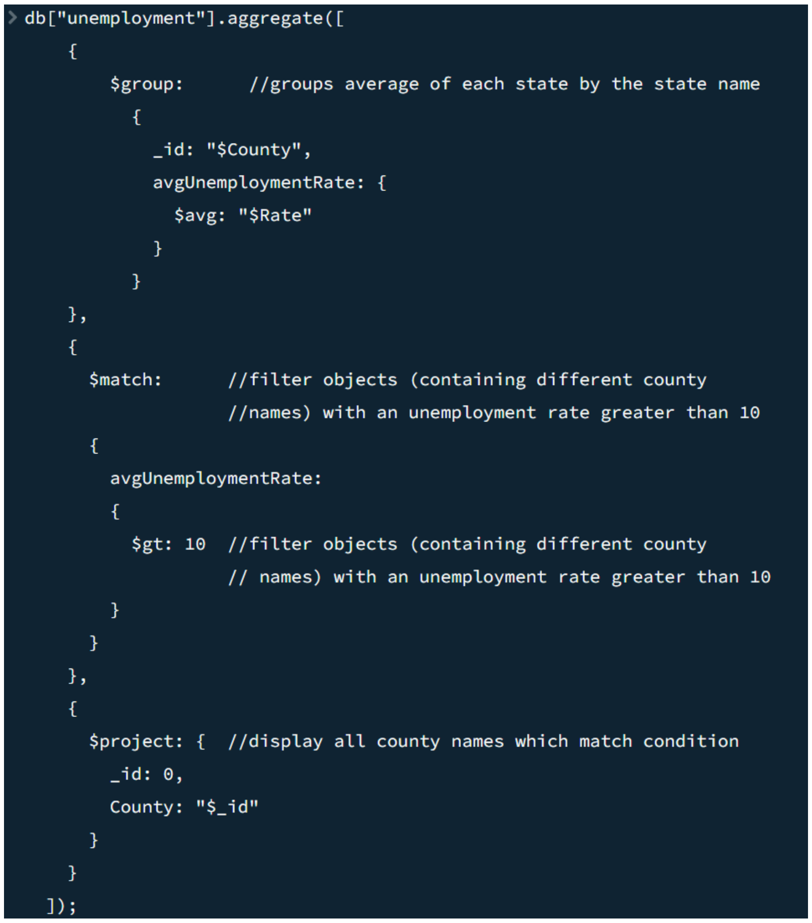
Query 3: What does this query compute?

The query db.unemployment.find({Rate : {$lt: 1.0}}).count() finds the number of documents having an employment rate of less than 1 percent. It uses the count function to count up objects which match those conditions.

Answer: 

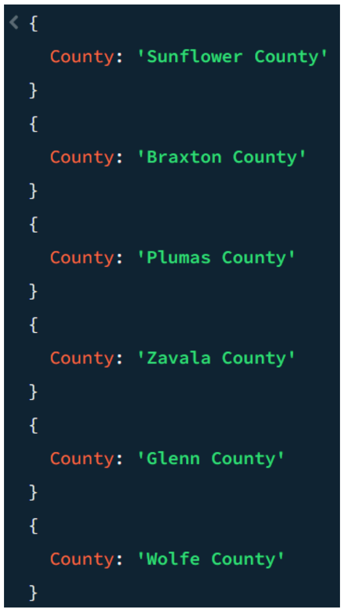
```
< 657
```

Query 4: Find all counties with unemployment rate higher than 10%



```
> db["unemployment"].aggregate([
    {
        $group:      //groups average of each state by the state name
          {
            _id: "$County",
            avgUnemploymentRate: {
              $avg: "$Rate"
            }
          }
    },
    {
      $match:      //filter objects (containing different county
                   //names) with an unemployment rate greater than 10
      {
        avgUnemploymentRate:
        {
          $gt: 10  //filter objects (containing different county
                   // names) with an unemployment rate greater than 10
        }
      }
    },
    {
      $project: {  //display all county names which match condition
        _id: 0,
        County: "$_id"
      }
    }
  ]);
```

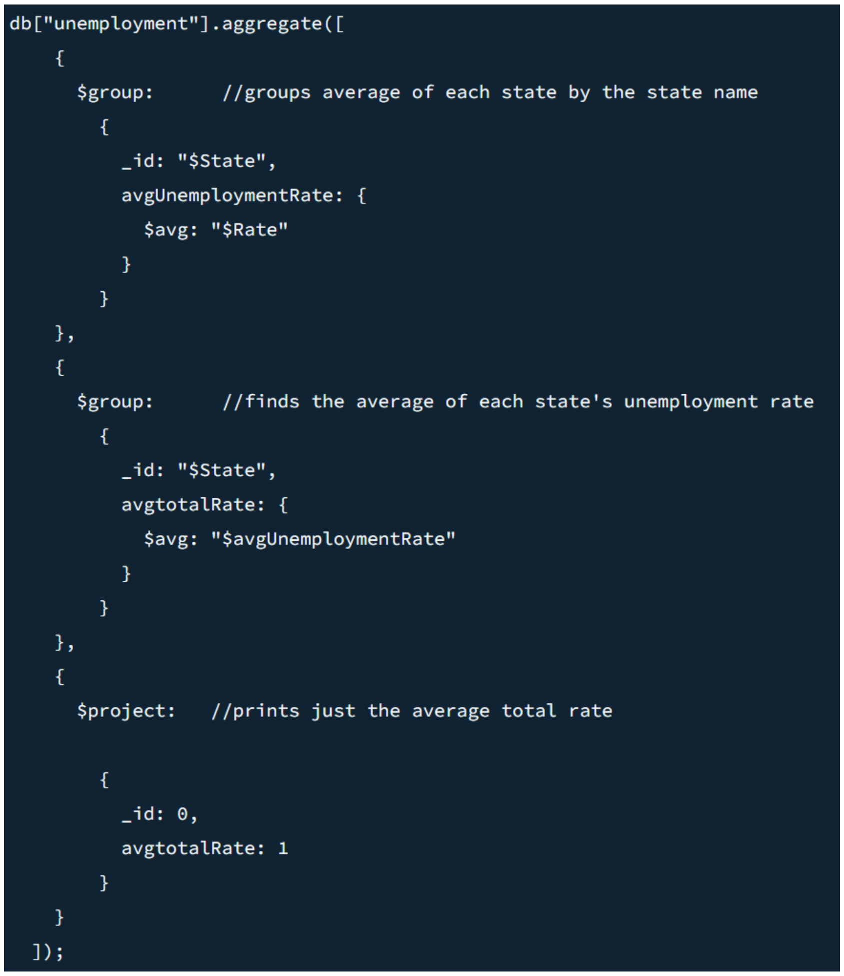
Sample output:



```
< {
    County: 'Sunflower County'
  }
  {
    County: 'Braxton County'
  }
  {
    County: 'Plumas County'
  }
  {
    County: 'Zavala County'
  }
  {
    County: 'Glenn County'
  }
  {
    County: 'Wolfe County'
  }
```

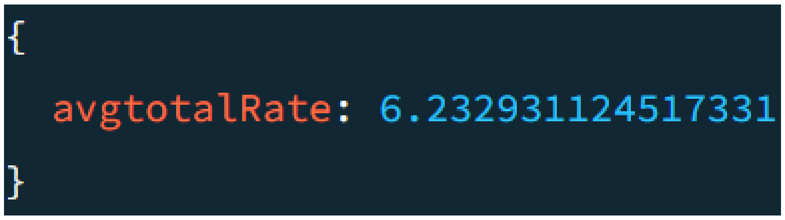
This query uses an aggregate function to group the results of stages together. The group stage groups the average of each county by its county name. The match stage filters results to only include those with rates higher than 10. The project function makes sure that only county names are outputted.

Query 5: Calculate the average unemployment rate across all states.



```
db["unemployment"].aggregate([
    {
      $group:      //groups average of each state by the state name
        {
          _id: "$State",
          avgUnemploymentRate: {
            $avg: "$Rate"
          }
        }
    },
    {
      $group:      //finds the average of each state's unemployment rate
        {
          _id: "$State",
          avgtotalRate: {
            $avg: "$avgUnemploymentRate"
          }
        }
    },
    {
      $project:   //prints just the average total rate

        {
          _id: 0,
          avgtotalRate: 1
        }
    }
  ]);
```

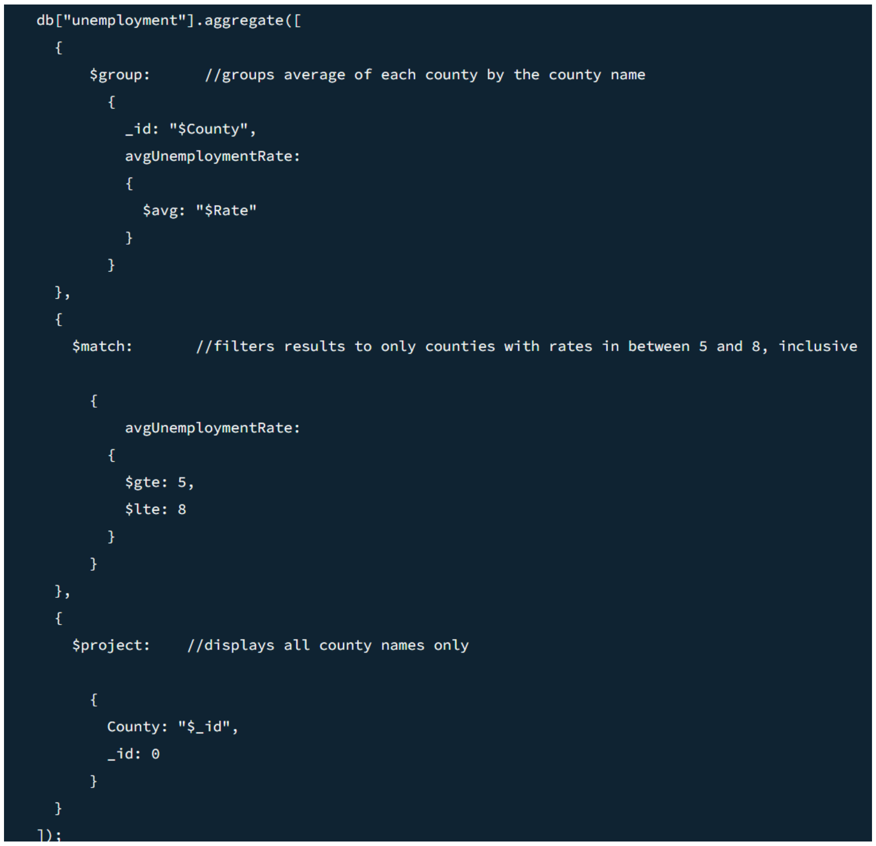
Output:



```
{
  avgtotalRate: 6.232931124517331
}
```

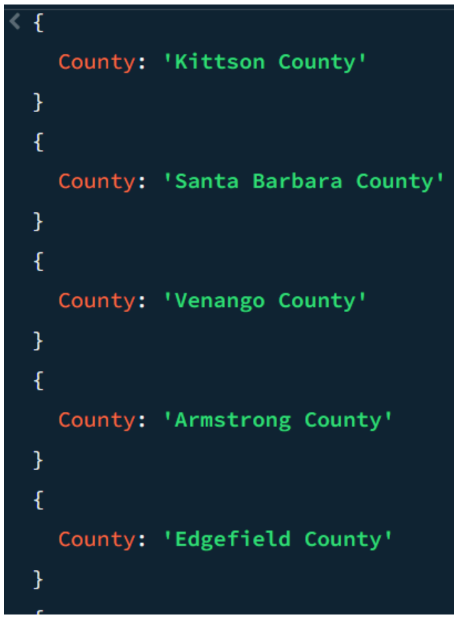
This query uses an aggregate function to group stage results. The first group function groups average unemployment rates based on each state. The second group function finds the average unemployment rate of the computed state averages. Finally, only the value of the average total rate is displayed.

Query 6: Find all counties with an unemployment rate between 5% and 8%.

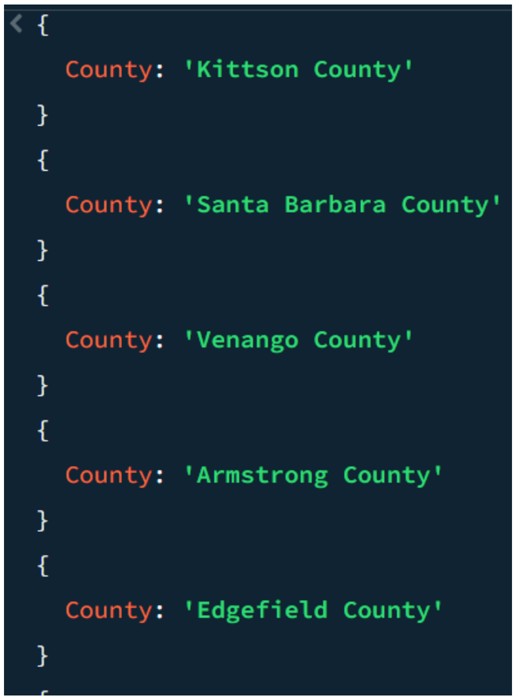


```
db["unemployment"].aggregate([
  {
      $group:      //groups average of each county by the county name
        {
          _id: "$County",
          avgUnemploymentRate:
          {
            $avg: "$Rate"
          }
        }
  },
  {
    $match:       //filters results to only counties with rates in between 5 and 8, inclusive

      {
          avgUnemploymentRate:
        {
          $gte: 5,
          $lte: 8
        }
      }
  },
  {
    $project:    //displays all county names only

      {
        County: "$_id",
        _id: 0
      }
  }
]);
```

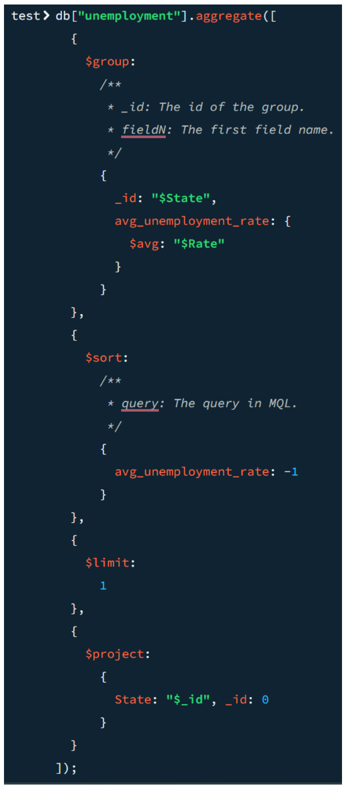
<u>**Sample Output:**</u>


```
{
  County: 'Kittson County'
}
{
  County: 'Santa Barbara County'
}
{
  County: 'Venango County'
}
{
  County: 'Armstrong County'
}
{
  County: 'Edgefield County'
}
```

This query contains a group stage which groups the average of each distinct county based on the county name. The match stage filters results with county rates that are in between 5 and 8, (assuming, inclusive). The project stage displays only county names which satisfy the condition.

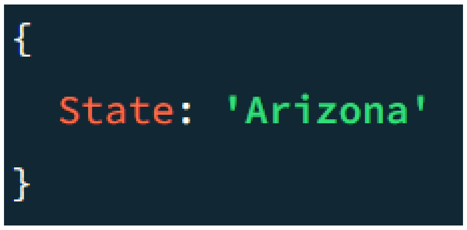

```
{
  County: 'Kittson County'
}
{
  County: 'Santa Barbara County'
}
{
  County: 'Venango County'
}
{
  County: 'Armstrong County'
}
{
  County: 'Edgefield County'
}
```

Query 7: Find the state with the highest unemployment rate. Hint. Use { $limit: 1 }



```
test> db["unemployment"].aggregate([
        {
          $group:
            /**
             * _id: The id of the group.
             * fieldN: The first field name.
             */
            {
              _id: "$State",
              avg_unemployment_rate: {
                $avg: "$Rate"
              }
            }
        },
        {
          $sort:
            /**
             * query: The query in MQL.
             */
            {
              avg_unemployment_rate: -1
            }
        },
        {
          $limit:
            1
        },
        {
          $project:
            {
              State: "$_id", _id: 0
            }
        }
      ]);
```

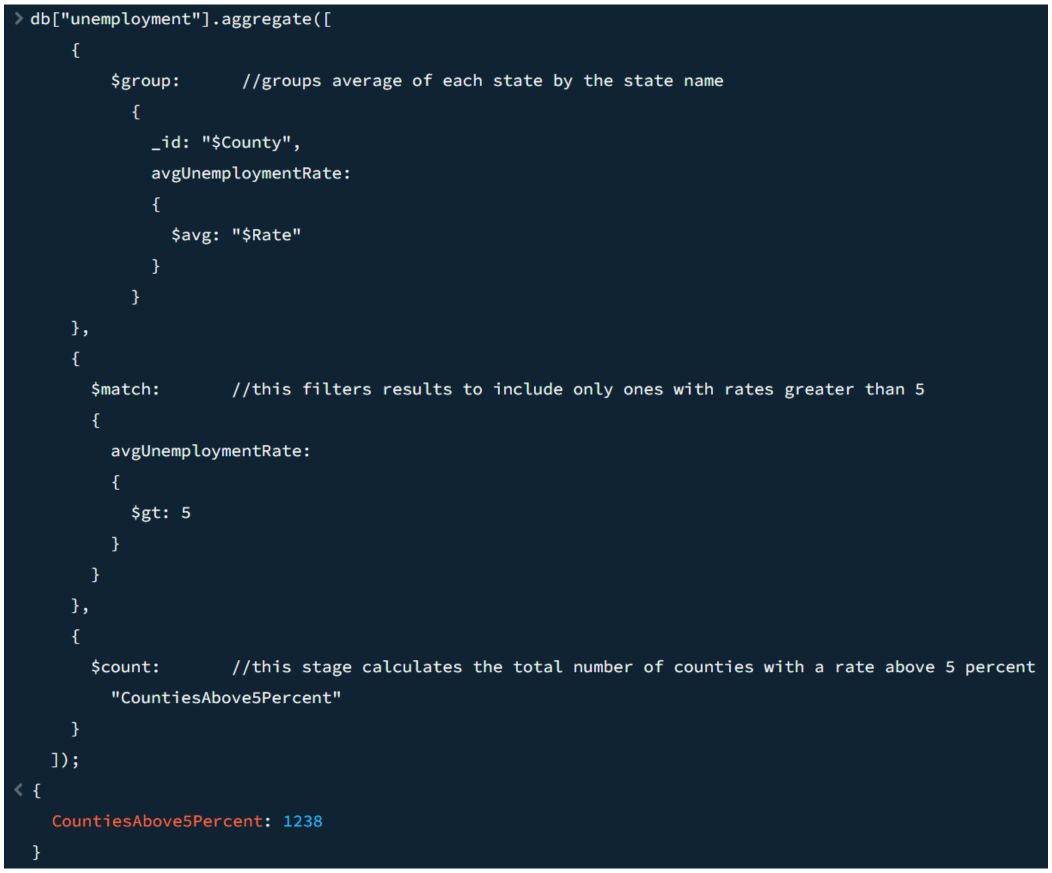
Output:



```
{
  State: 'Arizona'
}
```

This query once again finds the state's unemployment rate by calculating the average across all counties. It then sorts the unemployment rates in descending order. Finally, a limit 1 is used in order to display just the state with the highest average unemployment rate.

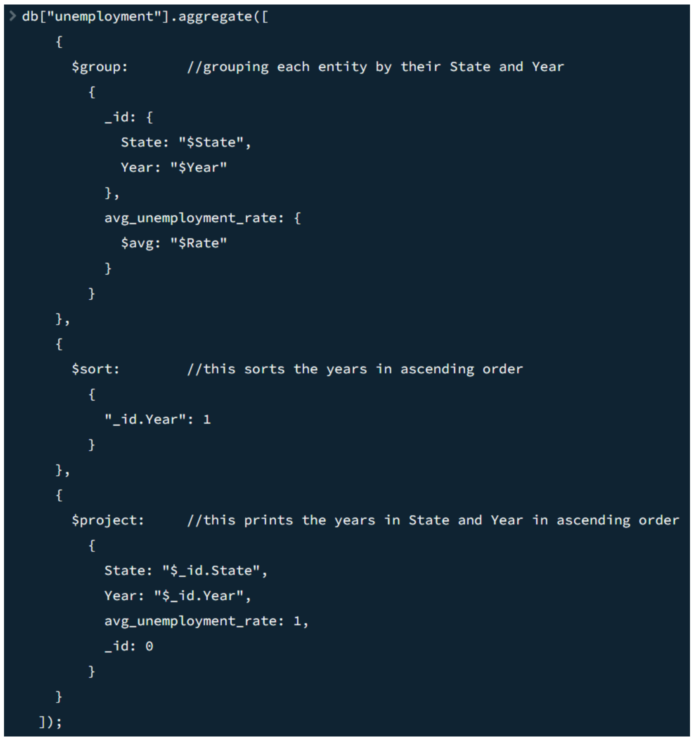
Query 8 (with output, on next page)



```
> db["unemployment"].aggregate([
      {
          $group:      //groups average of each state by the state name
            {
              _id: "$County",
              avgUnemploymentRate:
              {
                $avg: "$Rate"
              }
            }
      },
      {
        $match:       //this filters results to include only ones with rates greater than 5
        {
          avgUnemploymentRate:
          {
            $gt: 5
          }
        }
      },
      {
        $count:       //this stage calculates the total number of counties with a rate above 5 percent
          "CountiesAbove5Percent"
      }
    ]);
< {
    CountiesAbove5Percent: 1238
  }
```

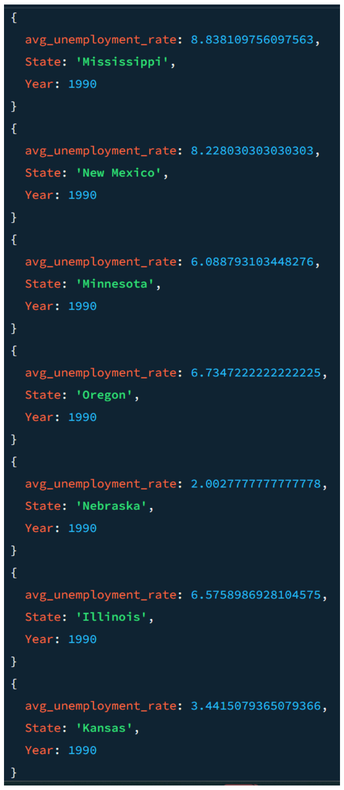
This query uses a group stage to group each distinct county's average by its name. The match stage to filter counties only having unemployment rates above 5 percent. It then uses a count stage which determines the number of those filtered documents, and then outputs the result.

Query 9: Calculate the average unemployment rate per state by year.



```
> db["unemployment"].aggregate([
    {
      $group:       //grouping each entity by their State and Year
        {
          _id: {
            State: "$State",
            Year: "$Year"
          },
          avg_unemployment_rate: {
            $avg: "$Rate"
          }
        }
    },
    {
      $sort:        //this sorts the years in ascending order
        {
          "_id.Year": 1
        }
    },
    {
      $project:     //this prints the years in State and Year in ascending order
        {
          State: "$_id.State",
          Year: "$_id.Year",
          avg_unemployment_rate: 1,
          _id: 0
        }
    }
  ]);
```

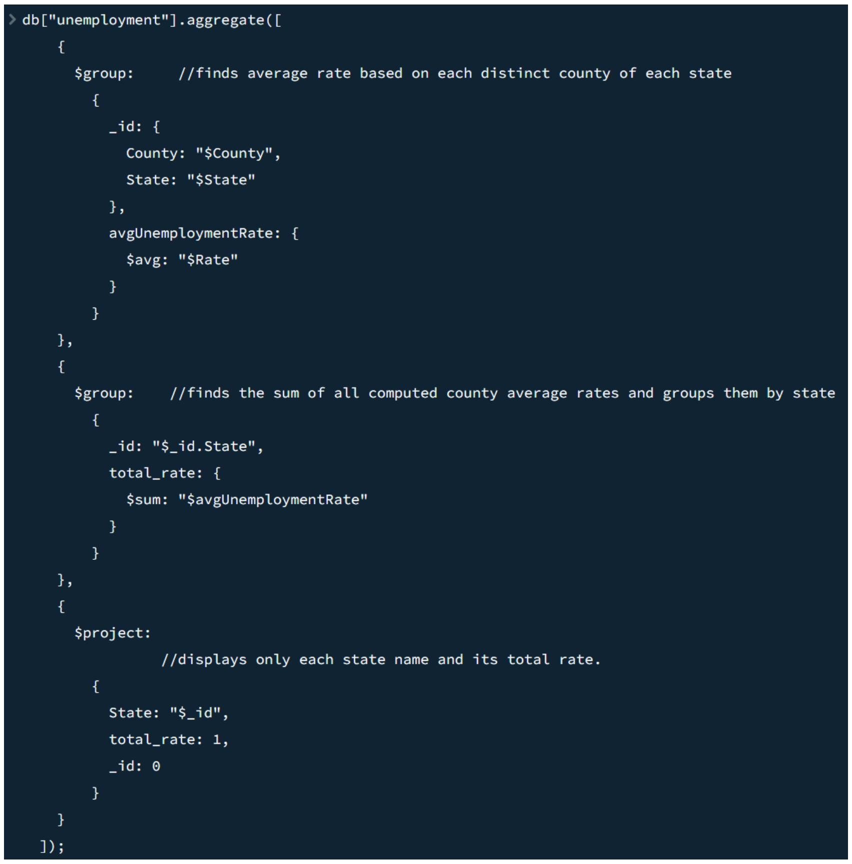
Output:



```
{
  avg_unemployment_rate: 8.838109756097563,
  State: 'Mississippi',
  Year: 1990
}
{
  avg_unemployment_rate: 8.228030303030303,
  State: 'New Mexico',
  Year: 1990
}
{
  avg_unemployment_rate: 6.088793103448276,
  State: 'Minnesota',
  Year: 1990
}
{
  avg_unemployment_rate: 6.7347222222222225,
  State: 'Oregon',
  Year: 1990
}
{
  avg_unemployment_rate: 2.0027777777777778,
  State: 'Nebraska',
  Year: 1990
}
{
  avg_unemployment_rate: 6.5758986928104575,
  State: 'Illinois',
  Year: 1990
}
{
  avg_unemployment_rate: 3.4415079365079366,
  State: 'Kansas',
  Year: 1990
}
```

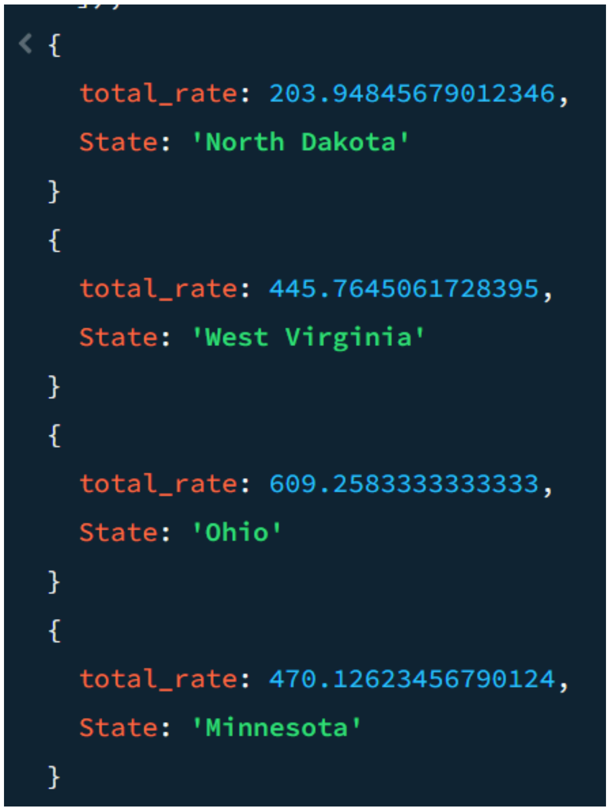
This query uses a group stage to group average unemployment rates for each state by their year. It then sorts the years in order (I added this step) for a clearer distinction on the years being grouped. It then displays those results using the project stage.

Extra Credit 1: For each state, calculate the total unemployment rate across all counties (sum of all county rates).



```
> db["unemployment"].aggregate([
    {
      $group:     //finds average rate based on each distinct county of each state
        {
          _id: {
            County: "$County",
            State: "$State"
          },
          avgUnemploymentRate: {
            $avg: "$Rate"
          }
        }
    },
    {
      $group:    //finds the sum of all computed county average rates and groups them by state
        {
          _id: "$_id.State",
          total_rate: {
            $sum: "$avgUnemploymentRate"
          }
        }
    },
    {
      $project:
              //displays only each state name and its total rate.
        {
          State: "$_id",
          total_rate: 1,
          _id: 0
        }
    }
  ]);
```

Sample Output:


```
< {
    total_rate: 203.94845679012346,
    State: 'North Dakota'
  }
  {
    total_rate: 445.7645061728395,
    State: 'West Virginia'
  }
  {
    total_rate: 609.2583333333333,
    State: 'Ohio'
  }
  {
    total_rate: 470.12623456790124,
    State: 'Minnesota'
  }
```

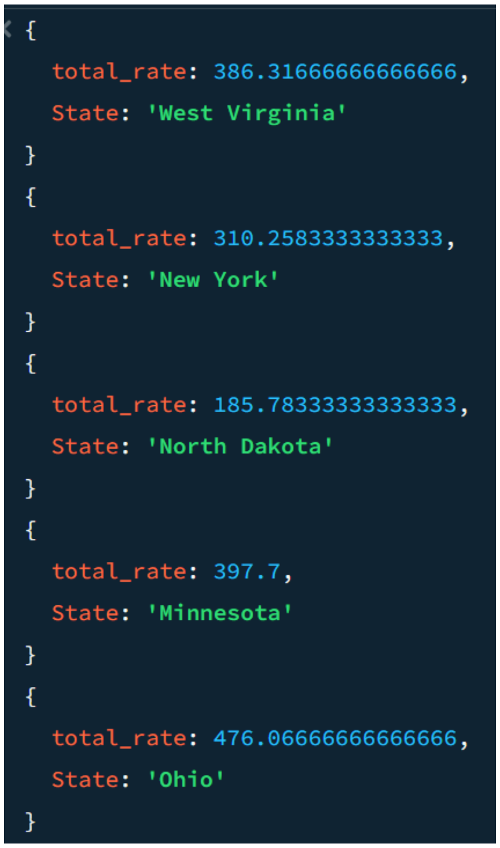
This query uses a group stage to find the average of each county's unemployment rate, for each state. Then, another group stage is used to find the sum of each state's distinct county rates (using the $sum function to accomplish this), and to organize the results by state. The project stage displays the total rates along with each state name.

Extra Credit 2: The same as Query 10 but for states with data from 2015 onward.

```
db["unemployment"].aggregate([
  {
    $match:   //filters documents which contain years from 2015 onward only.
    {
      Year: {$gte: 2015}
    }
  },
  {
   $group:     //finds average rate based on each distinct county of each state
    {
     _id: {
       County: "$County",
       State: "$State"
     },
     avgUnemploymentRate: {
       $avg: "$Rate"
     }
```

```
      }
    },
    {
      $group:    //finds the sum of all computed county average rates and groups them by state
        {
          _id: "$_id.State",
          total_rate: {
            $sum: "$avgUnemploymentRate"
          }
        }
    },
    {
      $project:
               //displays only each state name and its total rate.
        {
          State: "$_id",
          total_rate: 1,
          _id: 0
        }
    }
  ]);
```

Sample Output:



```
{
  total_rate: 386.31666666666666,
  State: 'West Virginia'
}
{
  total_rate: 310.2583333333333,
  State: 'New York'
}
{
  total_rate: 185.78333333333333,
  State: 'North Dakota'
}
{
  total_rate: 397.7,
  State: 'Minnesota'
}
{
  total_rate: 476.06666666666666,
  State: 'Ohio'
}
```

This query computes the total unemployment rate for each state just like the previous problem, except a match stage is used at the beginning to filter data containing years 2015 or greater only. This is why the total rates for each state are far less in this query than in the previous one.