For the first two weeks, we created the tables of job click data and job expiration date data on database. Then we generated basic descriptive statistics on the data available to us.

**DESCRIPTIVE STATISTICS ON CLICK DATA**

Using a 1 % random sample of the clickstream data, we examined both the day of week and time of day distribution of the clicks on job postings. We see that Monday - Thursday have relatively similar click counts, whereas Friday has a slightly lower click count, and Saturday/Sunday show even smaller click counts.

Looking at time of day, we see that clicks peak at 6 - 8 pm, presumably after working a long day at an unsatisfying job. Clicks steadily decline after that period until reaching a low point around 9 am, when they begin increasing again, but at a much faster rate than they declined, until the peak time period.

We generated an interactive graph of relationship between days before jobs expire and click time. The plot shows that as deadline approaches, there are more clicks on the job description, except in the last day click time will decrease a little bit.

We generated an interactive graph of relationship between number of clicks per day and the date. We found that there are generally more clicks on job descriptions when the date is later.

A histogram showing count of users grouped in bins based on their total # of clicks was created, showing a heavily right-skewed distribution. 35M users (~80%) only clicked on one posting. On the other end of the spectrum, 256 users clicked over 10k times.

Another histogram was created, displaying count of users grouped by their max # of clicks in one day. Again, this was highly right-skewed. Quite a few users had clicked over 1k times in a single day, drawing suspicions that they may be bots and not ‘real’ users.

The time each user had between their clicks was calculated, with the results being displayed in a histogram showing the count of total clicks grouped by the time spent by user until next click. The majority of times between clicks were measured to be <5 seconds or >60 minutes, but there were still a significant number between those times. These numbers may be able to be used as a proxy for how long the job was viewed or how long it took the applicant to apply to the job.

A metric we defined as User Click Streak (highest # of consecutive days that a specific user made at least one click) was calculated and visualized. Again, this was heavily right-skewed, with a maximum User Click Streak of 43 days. This metric may be useful in the future to identify which users are consistently visiting the site vs those who visit just once or twice.

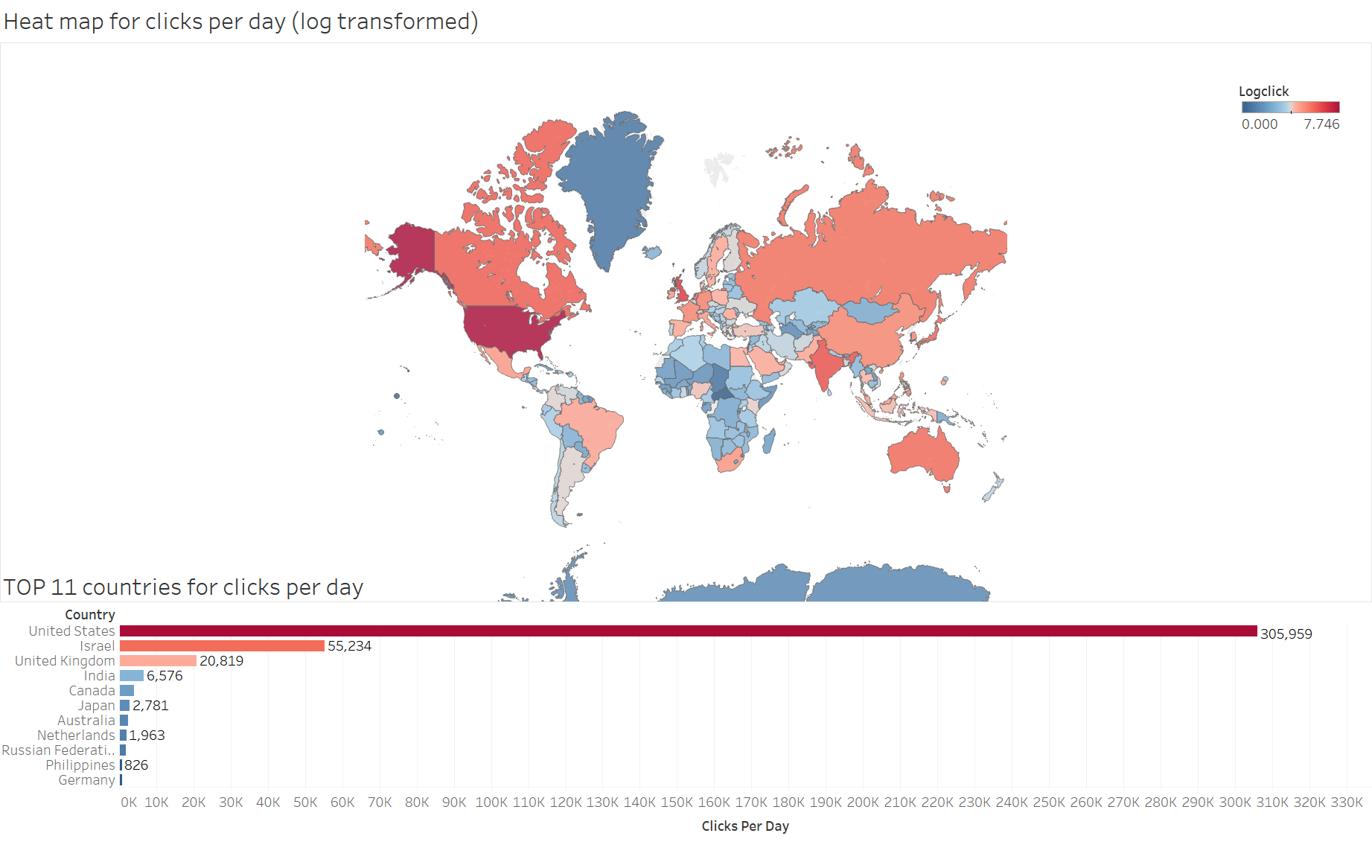
**IP TO LOCATION AND DESCRIPTIVE ANALYSIS**

(parsing with “logcombineparseip.py” in python)

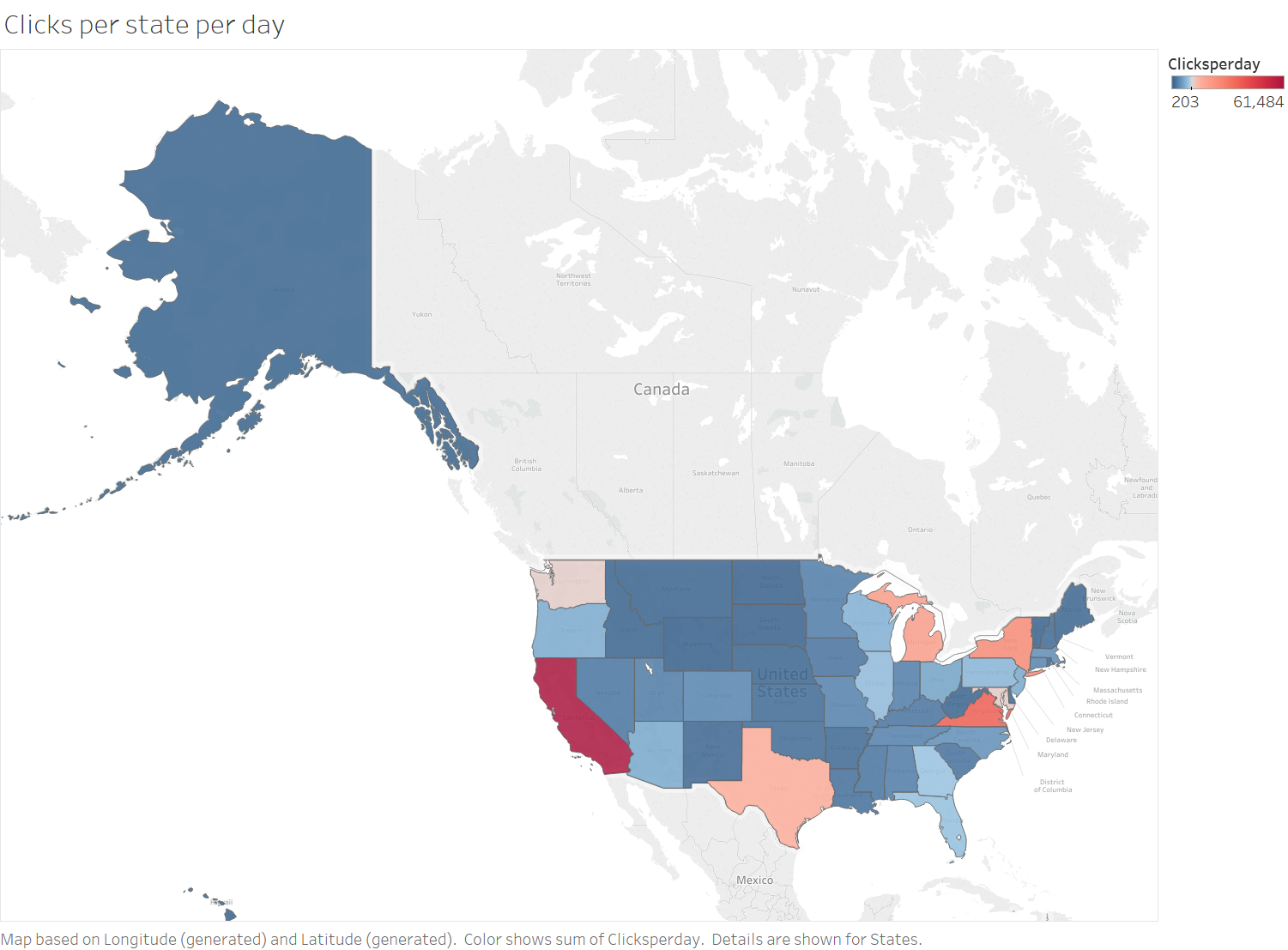
Given the scale of clickstream data we have, translating ip to physical address would be extremely time-consuming if we did it in SQL, or if for every ip address we perform a single query online via some IP2LOCATION API.

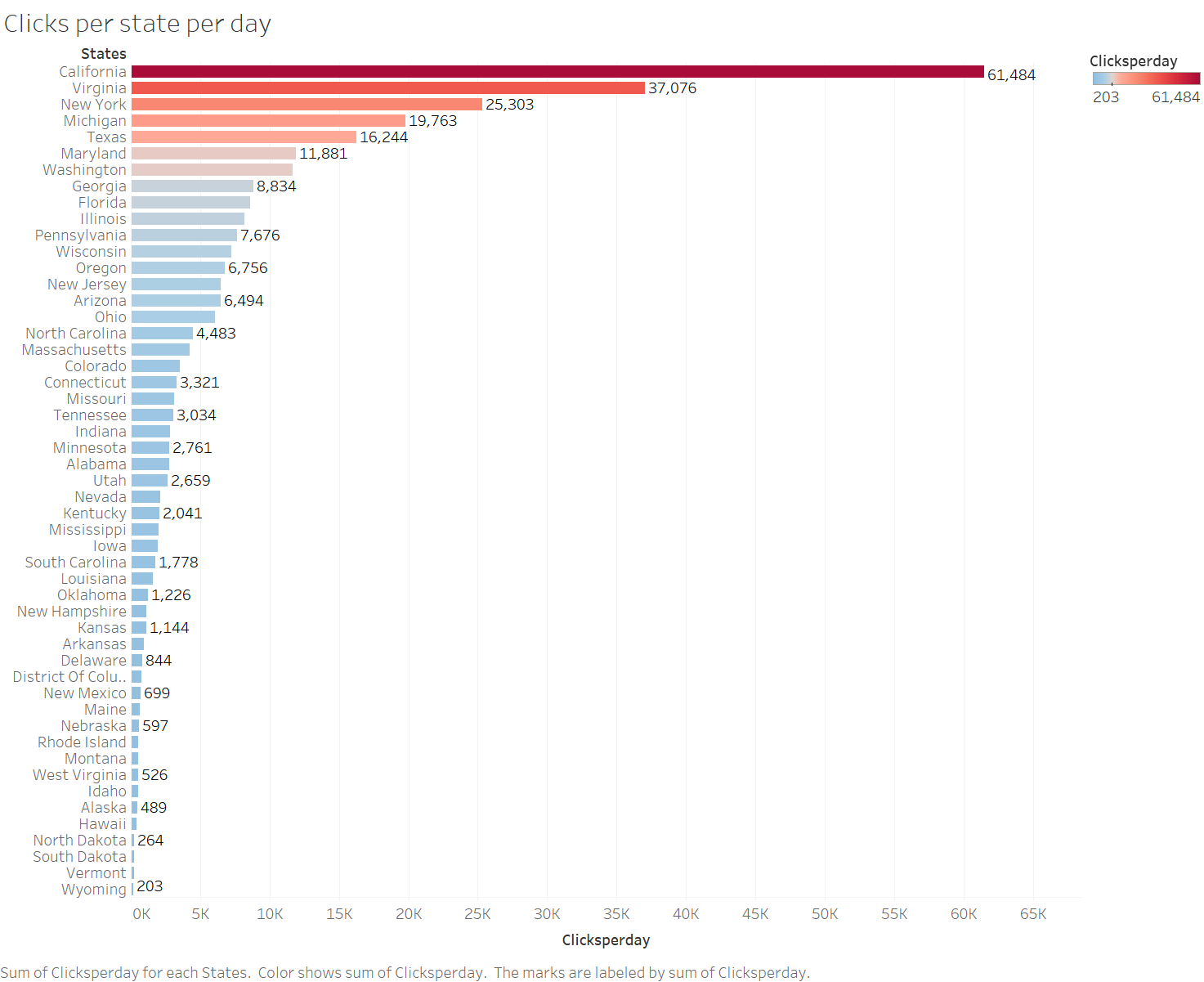
Therefore, we load a sorted IP2LOCATION database ('IP2LOCATION-LITE-DB9.csv') in python first. When concatenating the messy clickstream logs into a single, structured file, for every row that we successfully extracted an IP address via regex, we first translated the dot-decimal formatted IP address into decimal format, and then binary search can be performed to efficiently identify the ip-interval this very ip address belongs to. To reduce redundancy in storage, we do not extract the location details (country, region, city, etc.) directly. Instead, we store the start of the ip-interval (which works as a key in IP2LOCATION relation) this ip address belongs to. Once we need to query the location details for any click, we join these two tables, which is actually quite fast when provided this key.

Descriptive analysis are as follows. We are interested in that, aside from the United States, which countries are making job-query clicks, and how many clicks would they send per day. Below, we list the top 11 countries (USA + 10 other) and the details of their clicks per day.

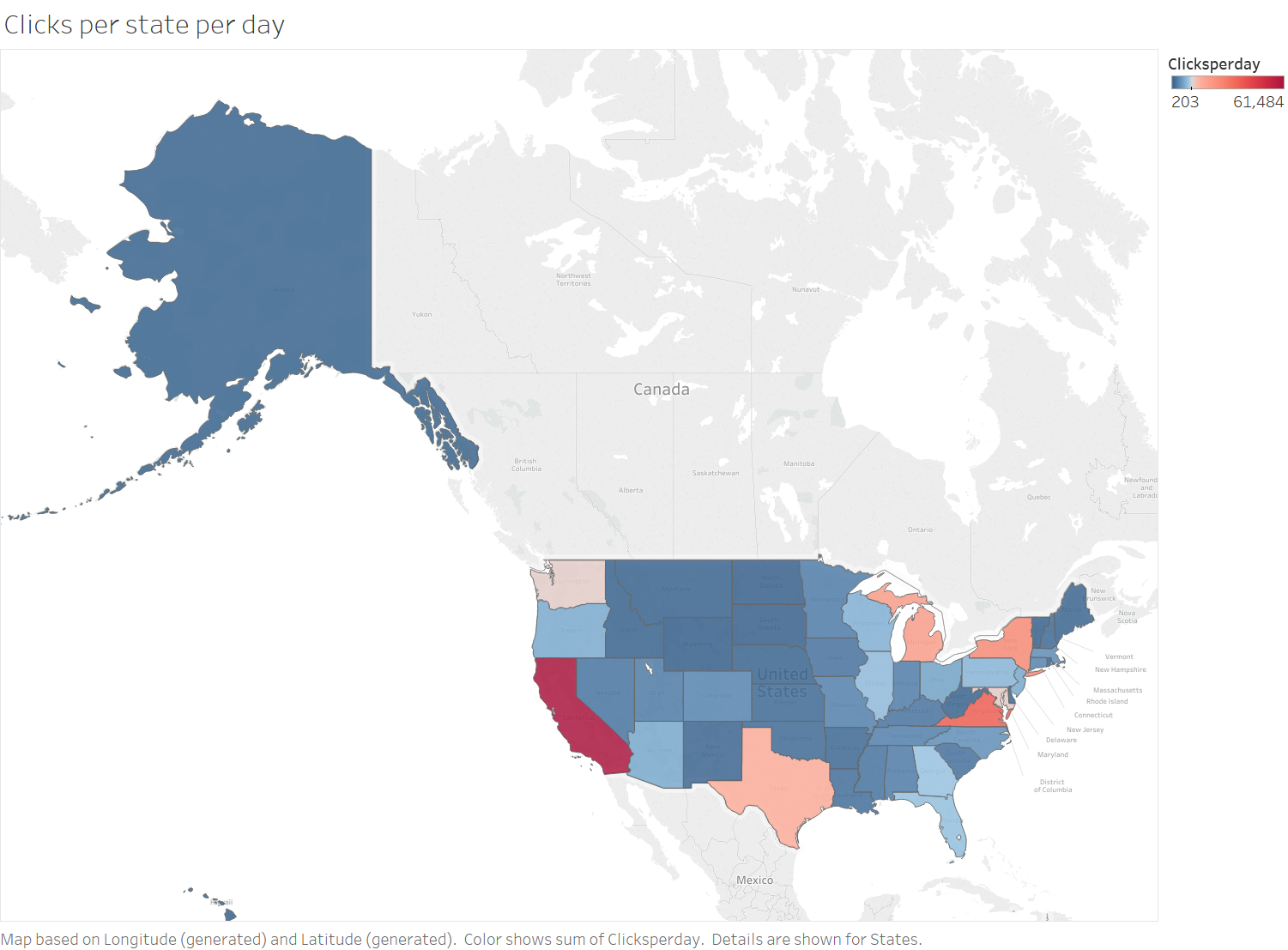


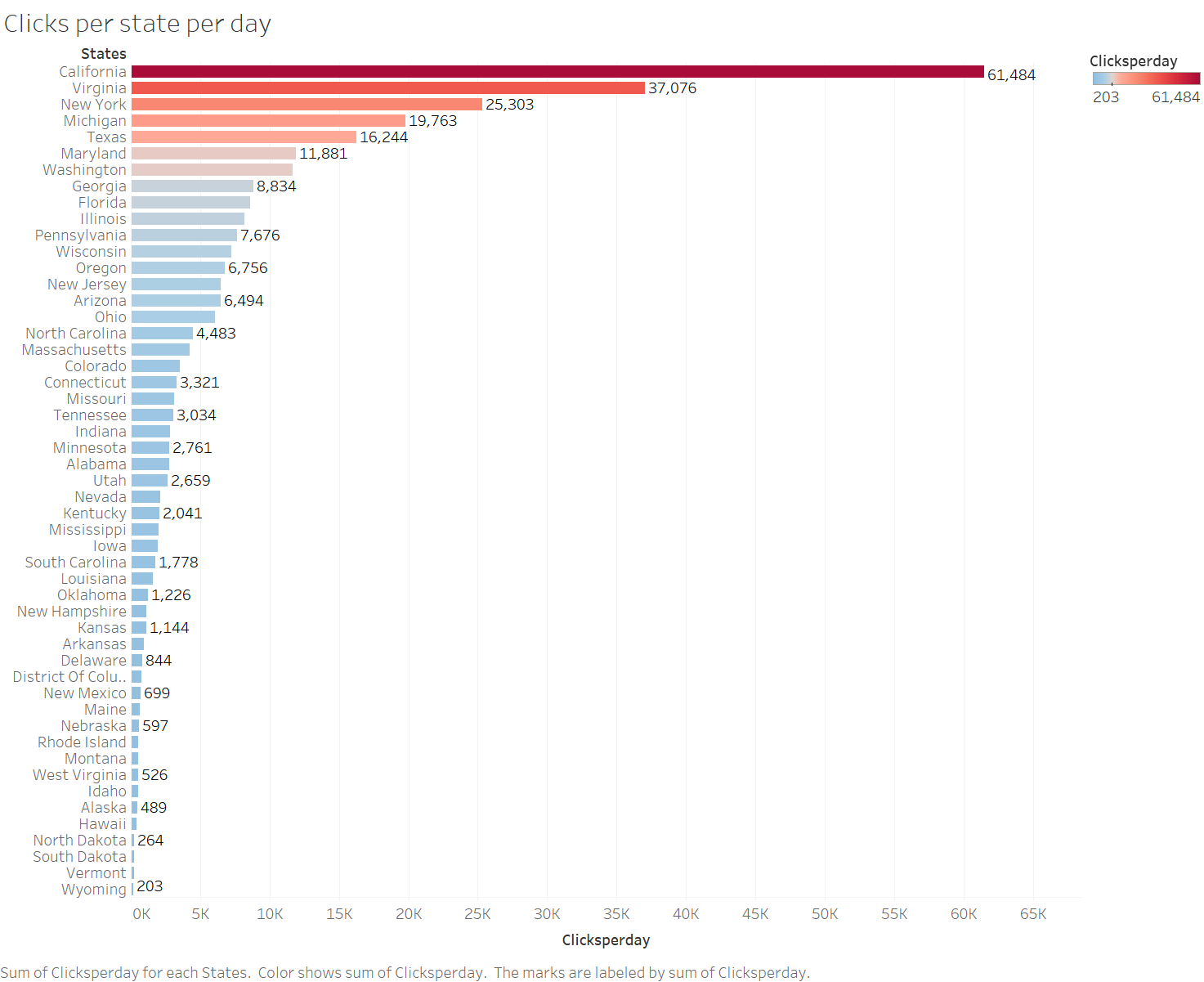
Then we take a closer look on the geographical distribution of clickstream in the USA. At a brief look, the # of clicks correlates significantly with population in state. That’s why we move on to the clicks per capita analysis





We combine the click counts with the population estimation from Wikipedia (<https://en.wikipedia.org/wiki/List_of_U.S._states_and_territories_by_population>), then a clear demonstration that people from which states are more interested in online job-hunting.





**Parsing job descriptions text**

In order to extract information from job descriptions text data, we parsed the texts with keywords matching, which results in the title of the job and job type (full-time, part-time, internship) of a certain job. The generated job title data has inaccuracy in it. For example, if the job’s title is manager, we are not sure if it’s a sales manager, or analytics manager, or a store manager. However, the accuracy of our algorithm is improving, and now 47% of all jobs have a certain job title.

Job type data is cleaner, and with simple data manipulation, we found that there are 1.9% internships in the job descriptions data, 5.5% part-time jobs and the rest are all full-time jobs.

Parsing the text data will enable us to add some new columns to the job descriptions table, so that we can perform more specific data analysis. The code to do the text analysis is description.py file on github.