Data normalization:

The original flat file was denormalized, in terms of having the most popular job skills and locations broken into separate columns. The advantage of denormalization was that it’s fast for processing, suitable for analytical purpose. For instance, we can just quickly do calculation on specific columns, like summing up the “R” or “Python” columns to see the percentage of jobs requiring these skills.

However, for this project, we needed to normalize our data. We created a star schema by separating the original file into 4 tables. They were “job\_post\_specific”, “job\_position”, “company” and “description” tables. Each of these tables was normalized and had a primary key that can join with a foreign key in other tables. For example, we can join the “job\_position” table with “company” table with the “company\_id”. We were aware that the original file is on job posting level – for example, Google is hiring a “data scientist” position in 4 different locations, and you see 4 entries in the data set. That is NOT a correct approach for this study.

As a result, we separated the original file into “job\_position” and “job\_post\_specific” levels. That means, using above Google as an example, you should only see 1 entry in the “job\_position” table, while we would see 4 entries in the “job\_post\_specific” table because Google is hiring the same position in 4 different locations. This distinction between the two tables is very important because if we count the number of jobs requiring a specific skill set, our numbers would get inflated if we query against the “job\_post\_specific” table. We also spent some time cleaning the data set and doing some simple change-data-capture, such as extracting the most recent company rating in the “company” table. The following analysis is based on unique “job\_position”.

Shiny:

The shiny app allows us to switch between “overall” and “industry”. We can look at the count of skills by company rating. Overall, Python and SQL are the most in demand, followed by Machine Learning and R. If we switch between industry, that won’t change much.

The top is almost always Python or SQL followed by R. It looks like being a data scientist, you should learn Python, SQL and R, regardless of which industry you are in.