Higthcharts: Gran flexibilidad, ofrece la posibilidad de realizar dashborads personalizados simplemente cogiendo los elementos de un json o añadiéndolos en el formato que desee el usuario. Es la biblioteca que nos ha ofrecido hasta la fecha mejores resultados en cuanto a adaptación de datos se refiere.

Charts.js: No ofrece tanta flexibilidad como high charts, tiene un formato de inclusión de datos único que no se puede modificar, y es poco flexible. La adaptación de los datos es prácticamente distinta en cada gráfica, lo que implica más líneas de código a la hora de implementarla.

d3.js: Ofrece una gran variedad de gráficas con las que trabajar, pero a nivel de código es poco flexible como charts.js. Además de eso está programado a muy bajo nivel, con lo que puede no ser cómodo para nuestros objetivos.

Jslate.js: Error a la hora de modificar a gusto el dashboard conjunto, es poco flexible en este aspecto, pero ofrece el mejor de los formatos quesito a la hora de representar por su flexibilidad, facilidad e intuición. Permite el filtrado de datos mediante un pequeño cuadro de texto, lo cual permite distintos análisis del usuario.

Dashing: Biblioteca muy original y muy llamativa. Permite la representación y seguimiento de datos en tiempo real aplicable al mundo de las finanzas, o a la programación en equipo a gran escala. En este caso la representación de los datos en cuanto a código refiere es bastante ambigua y poco intuitiva.

Dashku: Código abierto, pero no existen ejemplos ni referencias para tocar el código del autor. Habría que investigarla más.

dc.js: Biblioteca de gran flexibilidad aparente. Posee una gran comunidad de usuarios que han aportado su granito de arena y han creado sub-bibliotecas que pueden ser aplicadas a distintos ámbitos. Pese a todo la biblioteca no ofrece un código abierto claro más allá del código subido en github por el autor, lo que la hace al igual que con dashku una biblioteca difícil de abarcar en primera instancia.

Crossfilter: Al igual que dc.js se trata de una biblioteca que permite el filtrado de datos mediante ejemplos que están subidos, pero nuevamente no ofrece un apartado abierto en el que poder aprender sobre la biblioteca, lo que implica pegarse directamente con el código subido por el autor a github. Esta biblioteca está construida a partir de la anterior ya mencionada d3.

NVD3: Viendo el código y los ejemplos que vienen, y haciendo unas pequeñas pruebas con ella sobre el mismo código que ofrece en su página he podido ver que dicha biblioteca se asemeja mucho en uso a charts.js. Lo que ofrece básicamente es una biblioteca basada en d3, que saca de los ficheros json de manera directa su formato a la hora de aplicarla en el javascript, y que modifica su formato dependiendo de la gráfica que quiera usar.

Stashboard: Biblioteca que hace uso de python para subir, bajar y deletear datos. Su documentación es prácticamente nula salvo dos ejemplos sueltos situados en su página principal. Para ver en más en profundidad la biblioteca hay que pegarse directamente con el código del autor. En este caso la biblioteca a diferencia de otras es muy pequeña.

Grafana: Código abierto para descargar, pero carece de documentación que explique sus funciones. Lo único que posee es una pequeña aplicación de ejemplo dónde se pueden modificar unas gráficas ya existentes.

Freeboard: Código complejo solo disponible desde el repositorio fuente donde fue escrita. No tiene documentación suficiente, ejemplos o tutoriales.

Graylog2: Se trata de una biblioteca no abierta destinada al ámbito empresarial. No hay forma de estudiarla en primera instancia ni tampoco forma de descargarla.

Logstash: Biblioteca de alto nivel de código abierto pero nula documentación. La única forma de aprender sobre esta es meter mano directamente al código del autor el cual es altamente amplio.

Kibana: Utiliza una implementación semejante a logstash, pero de la misma forma que la anterior carece de ejemplos, tutoriales o documentación accesible para aprender sobre esta.