**Kibana基本使用**

2016年02月01日 17:39:31

* 102882

      现在你已经安装了Kibana，现在你一步步通过本教程快速获取Kibana核心功能的实践经验。学习完本教程，你将：

             1.加载案例数据到你安装的Elasticsearch中

             2. 定义至少一个索引匹配模式

             3.使用Discover功能探索你的数据

             4.建立一个visualization图形化地展示你的数据

             5.把许多visualization汇编组装成一个Dashboard

     本段内容假设你已经安装好了Kibana和Elasticsearch，并且Kibana连接到了Elasticsearch。

     视频教程也可获取

[High-level Kibana 4 introduction, pie charts](https://www.elastic.co/blog/kibana-4-video-tutorials-part-1)

[Data discovery, bar charts, and line charts](https://www.elastic.co/blog/kibana-4-video-tutorials-part-2)

[Tile maps](https://www.elastic.co/blog/kibana-4-video-tutorials-part-3)

[Embedding Kibana 4 visualizations](https://www.elastic.co/blog/kibana-4-video-tutorials-part-4)

**在你开始之前：加载案例数据**

     本段教程依赖如下数据集：

            1. 莎士比亚的所有著作，合适地解析成了各个字段：shakespeare.json。

            2. 随机生成的虚构账号数据：accounts.json

            3. 随机生成的日志文件：logs.jsonl

       以上数据可在[这里](http://download.csdn.net/detail/ming_311/9424993" \t "_blank)下载

      莎士比亚数据集由如下数据格式组织

**[javascript]** [view plain](https://blog.csdn.net/ming_311/article/details/50619859) [copy](https://blog.csdn.net/ming_311/article/details/50619859)

1. {
2. "line\_id": INT,
3. "play\_name": "String",
4. "speech\_number": INT,
5. "line\_number": "String",
6. "speaker": "String",
7. "text\_entry": "String",
8. }

账户数据集由如下数据格式组织

**[javascript]** [view plain](https://blog.csdn.net/ming_311/article/details/50619859) [copy](https://blog.csdn.net/ming_311/article/details/50619859)

1. {
2. "account\_number": INT,
3. "balance": INT,
4. "firstname": "String",
5. "lastname": "String",
6. "age": INT,
7. "gender": "M or F",
8. "address": "String",
9. "employer": "String",
10. "email": "String",
11. "city": "String",
12. "state": "String"
13. }

日志数据有几十个不同的字段，但是在教程中关注的字段如下：

**[javascript]** [view plain](https://blog.csdn.net/ming_311/article/details/50619859) [copy](https://blog.csdn.net/ming_311/article/details/50619859)

1. {
2. "memory": INT,
3. "geo.coordinates": "geo\_point"
4. "@timestamp": "date"
5. }

在导入莎士比亚数据集之前，我们需要为各个字段建立一个映射。映射把索引里的文档划分成逻辑组，指明字段的特征，如字段是否可被搜索、是否被标记、是否能被拆分成多个文字等。

使用以下命令为莎士比亚数据集建立一个映射。

**[javascript]** [view plain](https://blog.csdn.net/ming_311/article/details/50619859) [copy](https://blog.csdn.net/ming_311/article/details/50619859)

1. curl -XPUT http://localhost:9200/shakespeare -d '
2. {
3. "mappings" : {
4. "\_default\_" : {
5. "properties" : {
6. "speaker" : {"type": "string", "index" : "not\_analyzed" },
7. "play\_name" : {"type": "string", "index" : "not\_analyzed" },
8. "line\_id" : { "type" : "integer" },
9. "speech\_number" : { "type" : "integer" }
10. }
11. }
12. }
13. }
14. ';

这个映射指明了数据集的如下特征：

1. speaker字段是一个字符串，并且不被分析。这个字段的字符串被视为一个单元，即时字段值有多个文字。

2.play\_name同样符合上述特征。

3.line\_id和speech\_number是一个整数。

日志数据需要一个映射表明地理位置的经纬度，通过在那些字段使用一个geo\_point类型。

使用以下命令为日志数据建立一个geo\_point映射。

**[javascript]** [view plain](https://blog.csdn.net/ming_311/article/details/50619859) [copy](https://blog.csdn.net/ming_311/article/details/50619859)

1. curl -XPUT http://localhost:9200/logstash-2015.05.18 -d '
2. {
3. "mappings": {
4. "log": {
5. "properties": {
6. "geo": {
7. "properties": {
8. "coordinates": {
9. "type": "geo\_point"
10. }
11. }
12. }
13. }
14. }
15. }
16. }
17. ';

**[javascript]** [view plain](https://blog.csdn.net/ming_311/article/details/50619859) [copy](https://blog.csdn.net/ming_311/article/details/50619859)

1. curl -XPUT http://localhost:9200/logstash-2015.05.19 -d '
2. {
3. "mappings": {
4. "log": {
5. "properties": {
6. "geo": {
7. "properties": {
8. "coordinates": {
9. "type": "geo\_point"
10. }
11. }
12. }
13. }
14. }
15. }
16. }
17. ';

**[javascript]** [view plain](https://blog.csdn.net/ming_311/article/details/50619859) [copy](https://blog.csdn.net/ming_311/article/details/50619859)

1. curl -XPUT http://localhost:9200/logstash-2015.05.20 -d '
2. {
3. "mappings": {
4. "log": {
5. "properties": {
6. "geo": {
7. "properties": {
8. "coordinates": {
9. "type": "geo\_point"
10. }
11. }
12. }
13. }
14. }
15. }
16. }
17. ';

那些账号数据不需要任何映射，所以这个时候我们使用Elasticsearch的批量导入API输入数据，使用如下命令：

**[javascript]** [view plain](https://blog.csdn.net/ming_311/article/details/50619859) [copy](https://blog.csdn.net/ming_311/article/details/50619859)

1. curl -XPOST 'localhost:9200/bank/account/\_bulk?pretty' --data-binary @accounts.json
2. curl -XPOST 'localhost:9200/shakespeare/\_bulk?pretty' --data-binary @shakespeare.json
3. curl -XPOST 'localhost:9200/\_bulk?pretty' --data-binary @logs.jsonl

这些命令将会花费一段时间来执行，视可利用计算资源而定。

使用如下命令验证成功导入：

**[javascript]** [view plain](https://blog.csdn.net/ming_311/article/details/50619859) [copy](https://blog.csdn.net/ming_311/article/details/50619859)

1. curl 'localhost:9200/\_cat/indices?v'

你将会看到输出如下类似的信息

**[javascript]** [view plain](https://blog.csdn.net/ming_311/article/details/50619859) [copy](https://blog.csdn.net/ming_311/article/details/50619859)

1. health status index               pri rep docs.count docs.deleted store.size pri.store.size
2. yellow open   bank                  5   1       1000            0    418.2kb        418.2kb
3. yellow open   shakespeare           5   1     111396            0     17.6mb         17.6mb
4. yellow open   logstash-2015.05.18   5   1       4631            0     15.6mb         15.6mb
5. yellow open   logstash-2015.05.19   5   1       4624            0     15.7mb         15.7mb
6. yellow open   logstash-2015.05.20   5   1       4750            0     16.4mb         16.4mb

**1. 定义你的索引模式匹配**

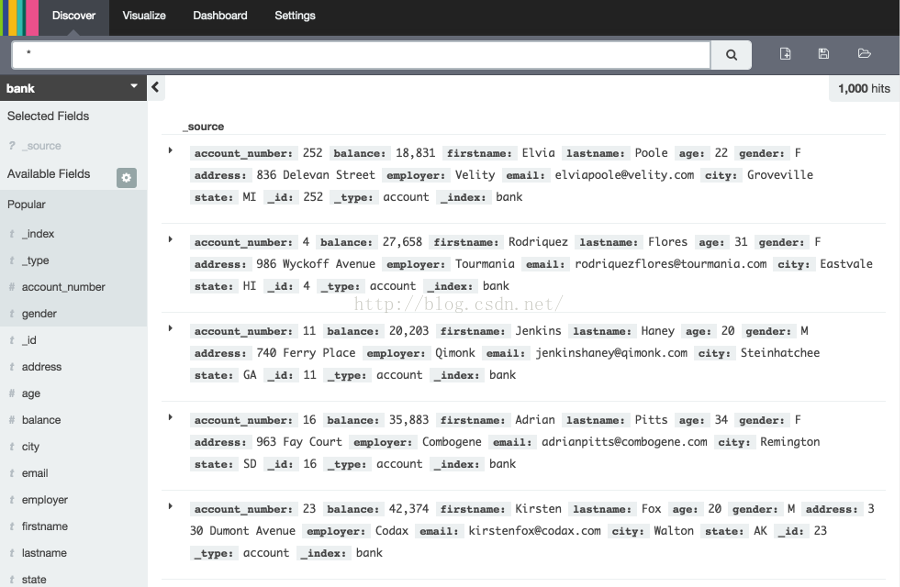
       每一个数据集导入到Elasticsearch后会有一个索引匹配模式，在上段内容莎士比亚数据集有一个索引名称为shakespeare，账户数据集的索引名称为bank。一个索引匹配模式就是一个字符串包含可选的通配符，它能匹配多个索引。比如，在常用的日志案例中，一个典型的索引名称包含MM-DD-YYYY格式的日期，因此一个5月的索引匹配模式可能是这样：logstash-2015.05\*。

      在本教程中任何匹配模式匹配到我们导入的索引将会起作用。打开一个浏览器，访问localhostL:5601，点击Setting页面，然后点击indices标签，点击Add New按钮，定义一个新的模式匹配。两个数据集——莎士比亚剧和虚构的账号不包含时间系列的数据，确保Index contains time-based events复选框没有勾选，当你在为这些数据集创建模式匹配的时候。指定shaks\*为莎士比亚数据集的一个模式匹配，然后点击Create按钮定义一个模式匹配，接着定义一个ba\*的模式匹配。

    Logstash数据集包含时间系列的数据，所以在点击Add New按钮创建完模式匹配后，确保Index contains time-based events复选框勾选，并在Time-field name下拉列表中选择@timestamp字段。

**发现你的数据**

   点击Discover页面展示Kibana的数据发现功能。



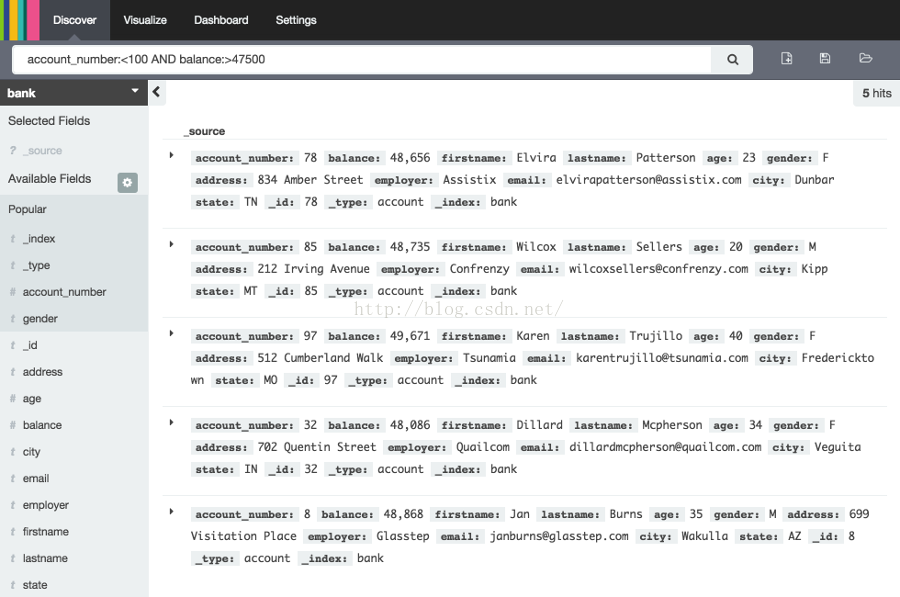
在标签页面的正下方，有一个查询框用于搜索你的数据。搜索需要一个特定的查询语法，它们能让你创建自己的搜索，点击查询框右边的按钮能保存这些搜索。在查询框的下方，当前的索引匹配模式显示在一个下拉选中，选择下拉选以改变匹配模式。你能用字段名和你感兴趣的值构建一个搜索，数字类型的数据可使用比较操作符比如>、<、=等,你可使用AND、OR、 NOT逻辑符连接元素，必须是大写。

    试着选择ba\*模式匹配，然后把下面的查询放到查询框

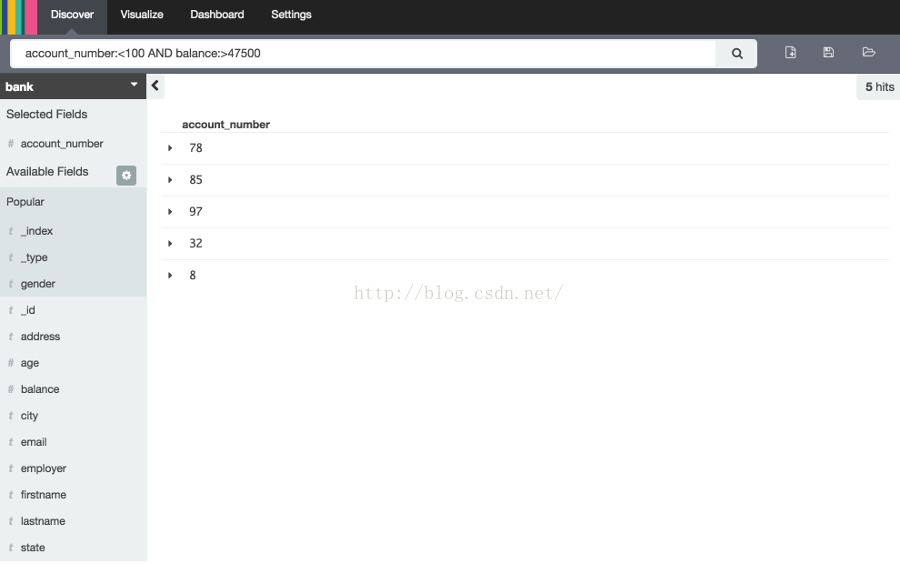
**[javascript]** [view plain](https://blog.csdn.net/ming_311/article/details/50619859) [copy](https://blog.csdn.net/ming_311/article/details/50619859)

1. account\_number:<100 AND balance:>47500

搜索返回所有账户号码0到99且薪水超过4,7500的账户。如果你使用相关的案例数据，将会返回5条数据：账户号码8，32,78,85和97。



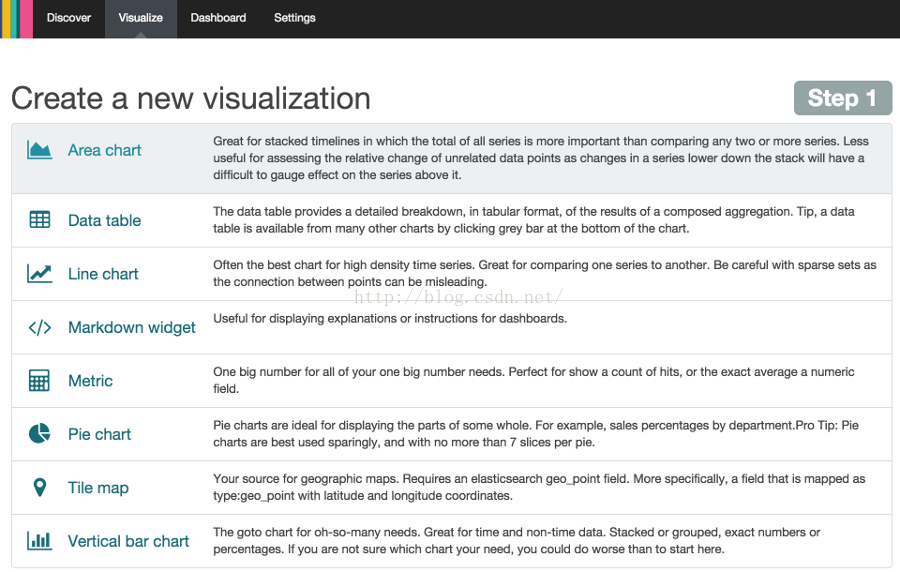
为了窄化显示某些感兴趣的字段，高亮索引模式匹配下面的列表中的字段，然后点击Add按钮。在这个例子中，注意怎么实现的，添加一个account\_number字段后改变界面显示从5条记录的完整文本到一个只有账户号码的简单列表。



**2. 数据可视化：不只是发现数据**

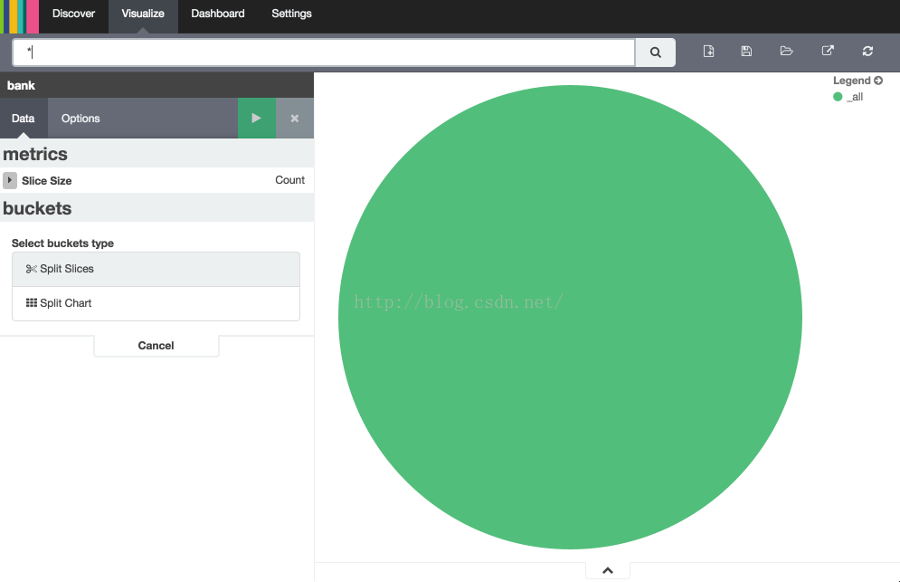
Visualize页面的可视化工具能使你用好几种不同的方式展示你数据集的很多方面。

点击Visualize页面开始



点击Pie chart，然后点击from a new search，选择ba\*索引匹配模式。

可视化依赖Elasticsearch聚合的两种类型：量聚合和刻度聚合。量聚合根据你指定的标准整理数据，比如，在我们的账户数据集中，我们可以建立一个账户薪水的范围，然后显示落在每个薪水范围的总比率。界面显示出一个完整的饼，因为我们现在还没有指定任何量值。

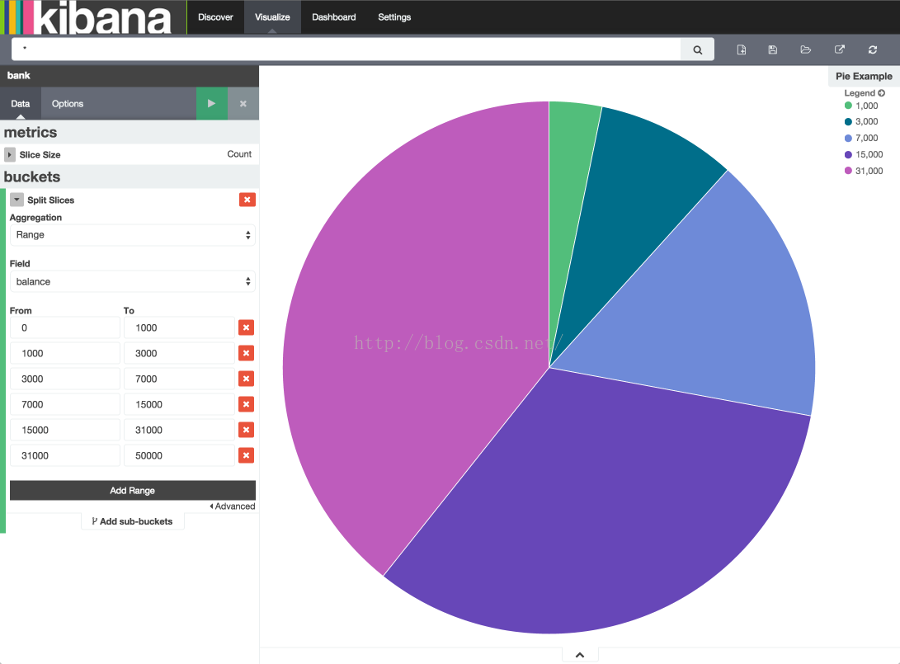


在Select buckets type下拉列表中，选择Split Slices，然后在Aggregation下拉列表中选择Range选项，在字段下拉列表中选择balance字段，点击Add Range按钮4次把区间增加到6个，输入一下区间。

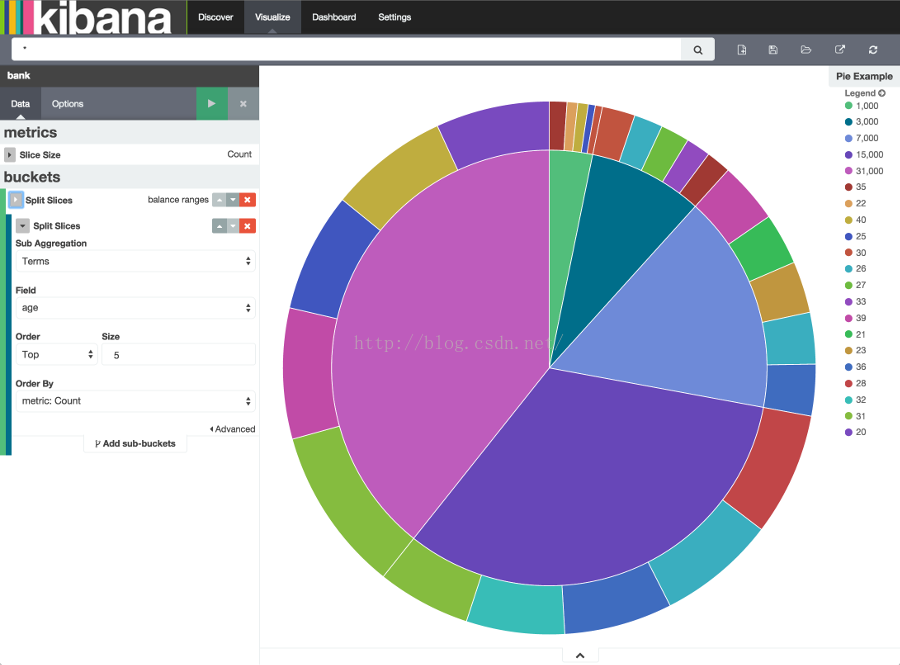
**[javascript]** [view plain](https://blog.csdn.net/ming_311/article/details/50619859) [copy](https://blog.csdn.net/ming_311/article/details/50619859)

1. 0             999
2. 1000         2999
3. 3000         6999
4. 7000        14999
5. 15000       30999
6. 31000       50000

点击应用按钮https://img-blog.csdn.net/20160203111227371?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQv/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center显示图表。

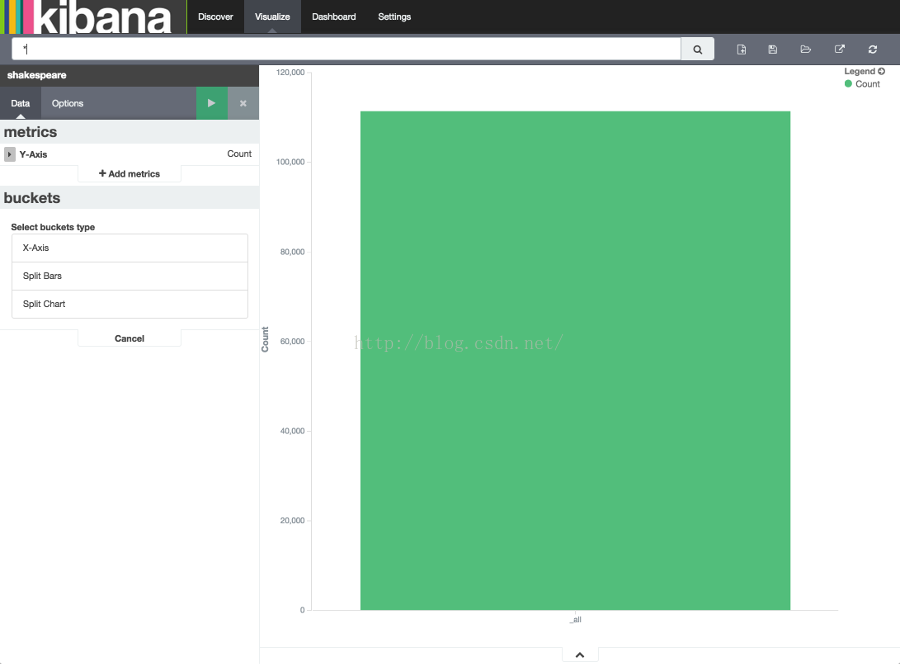


图表显示出这1000个账户落在不同薪水范围的比率。为了看数据的另一个维度，我们要添加另一个量聚合。我们可以更进一步地把每个区间依据薪水所有者的年龄拆分。在下面点击Add sub-buckets，然后点击Split Slice，在下拉选中选择Terms选项和age字段，点击绿色Apply changes按钮https://img-blog.csdn.net/20160203111227371?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQv/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center在新的结果中增加一个外环。

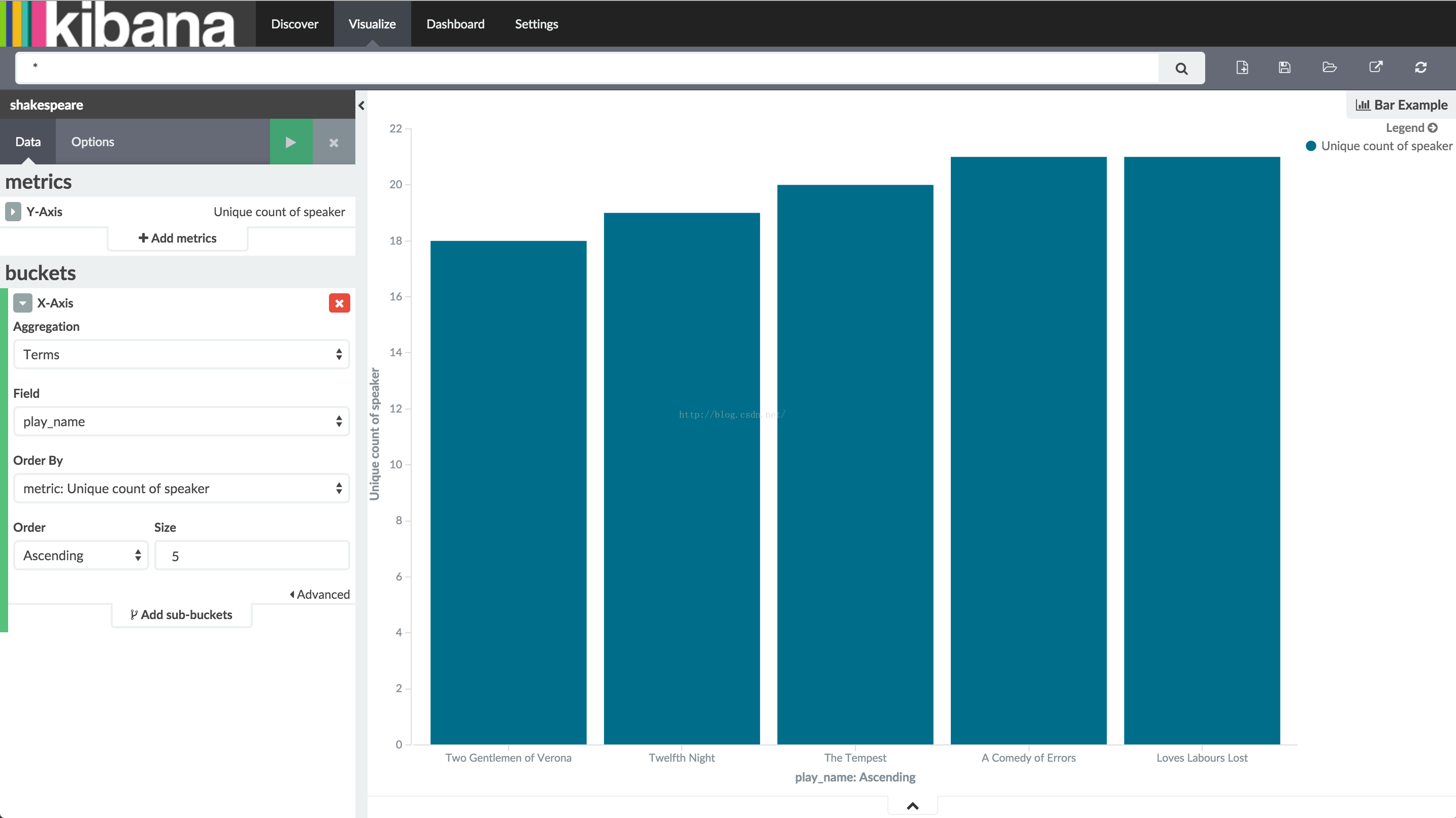


通过点击搜索字段的右边的按钮保存图表，把这个图表命名为Pie Example。

下一步，我们打算制作一个条形图。点击New Visualization按钮，然后点击Vertical bar chart。选择From a new search，然后选定shakes\*模式匹配。你将会看到单个大条形图，因为到现在为止我们还没有定义任何量值。

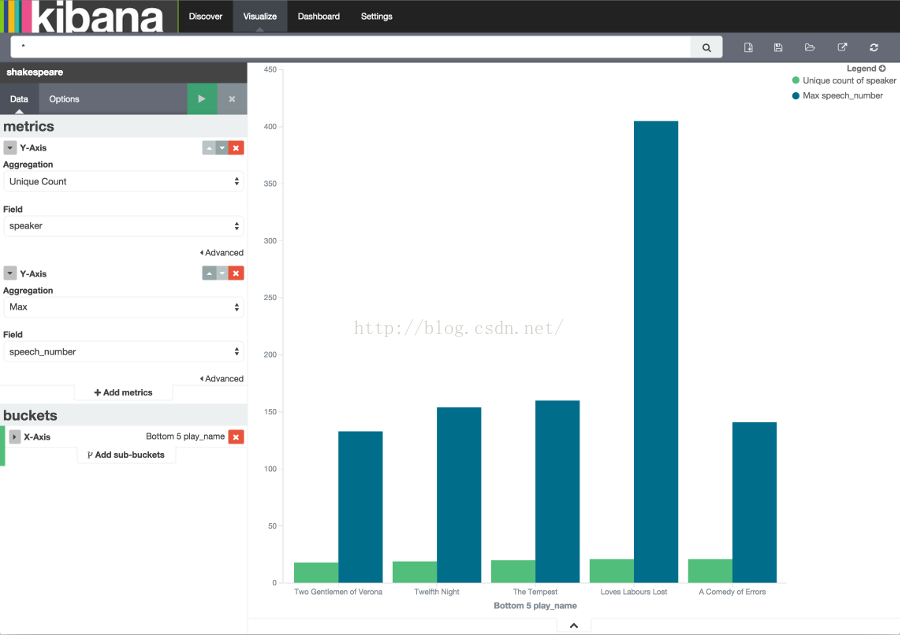


对于Y轴的刻度聚合，选择speaker作为Unique Count的字段。对于莎士比亚戏剧，知道那部戏剧需要最少数量的台前幕后人员可能是很有用的，如果你的戏剧公司短缺演员的话。对于X轴的量值，选择Terms聚合和play\_name字段。对于排序，选择Ascending，Size保持默认值5。让其他参数保持默认值，然后点击Apply cganges按钮https://img-blog.csdn.net/20160203111227371?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQv/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center，你的图表应该看起来像下面那样。



       注意一下每部剧名是怎么显示成一个完整的词组而不是被拆分成单独的单词。这是我们在教程的前段部分设置映射的结果，我们把play\_name标记为 not analyzed。鼠标移到每一个条上以tooltip形式显示每个剧台前幕后的数量。你可以关掉这个显示方式或者改变你图表的其他选项通过点击左上方的Option标签。

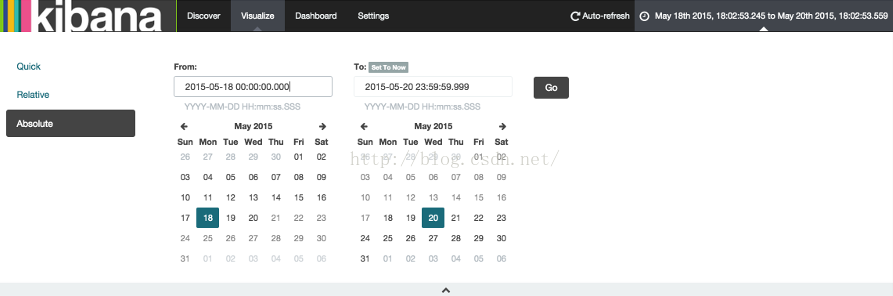
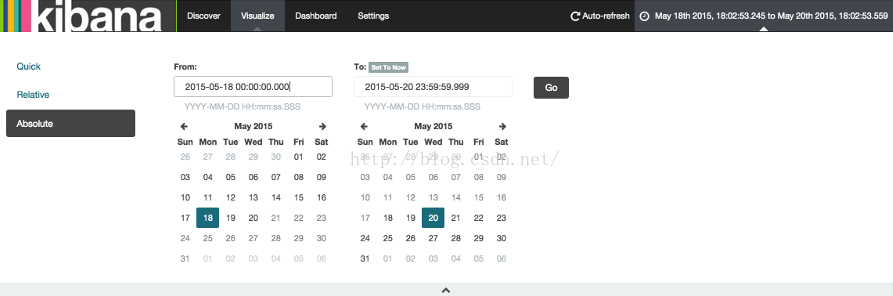
      既然你有了莎士比亚剧中最小的演员表，你可能感兴趣知道这些剧本中哪一个对单个演员的要求最高，通过显示给定剧情的最大对话量。用Add metrics按钮增加一个Y轴聚合，为speech\_number选择Max聚合。在option页面选择Bar Mode的grouped选项，然后点击Apply changes按钮https://img-blog.csdn.net/20160203111227371?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQv/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center，你的图表现在看起来应该是这样：



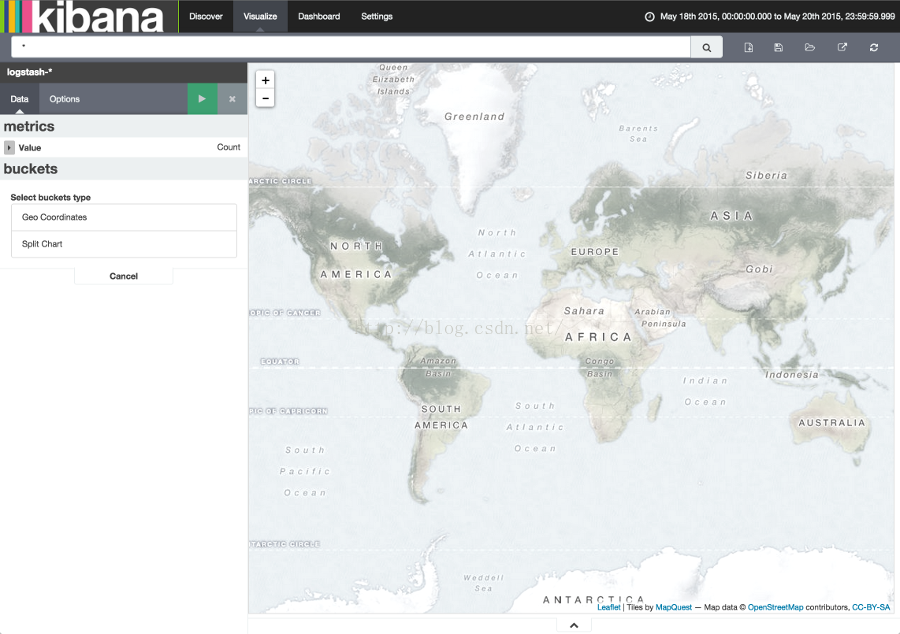
正如你所看到的，和其他剧比较起来，《Love’s Labours Lost》经常有最高数量的对话，因此和能会对一个演员的记忆要求更高。

保存图表的名称为Bar Example。

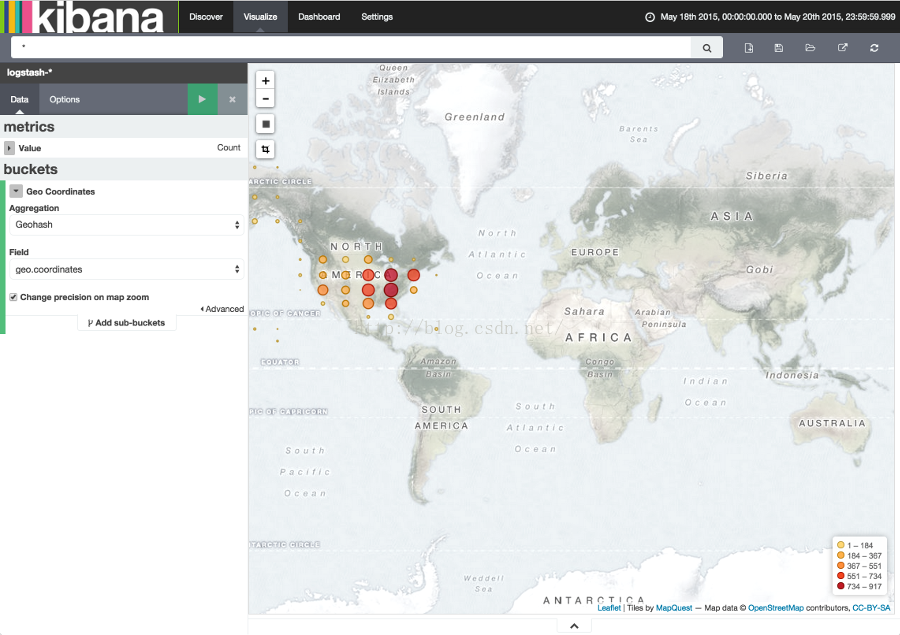
接下来，我们要制作一个瓦片地图来可视化一些地理数据。点击**New Visualization**按钮，然后点击**Tile map**按钮，选择from a new search和logstash-\*模式匹配。在Kinaba界面的右上方的时间选择器为我们的要发掘的数据定义一个时间窗口，点击**Absolute**按钮，然后设置起始时间为2015-5-18结束时间到2015-5-20。



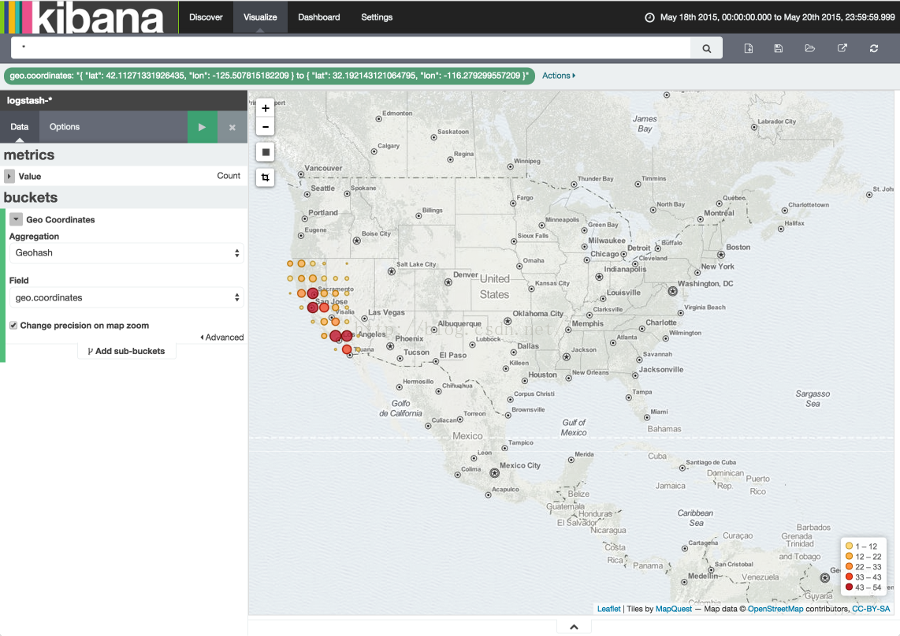
一旦我们设置好了时间区间，点击Go按钮，然后点击底部小小的向上箭头关闭时间选择器。你可以看到整个世界的地图，因为我们现在还没有定义任何量值。



选择**Geo Coordinates**作为量值，然后点击那个绿色的Apply changes按钮https://img-blog.csdn.net/20160203111227371?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQv/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center，你的图表现在看起来应该是这样的：

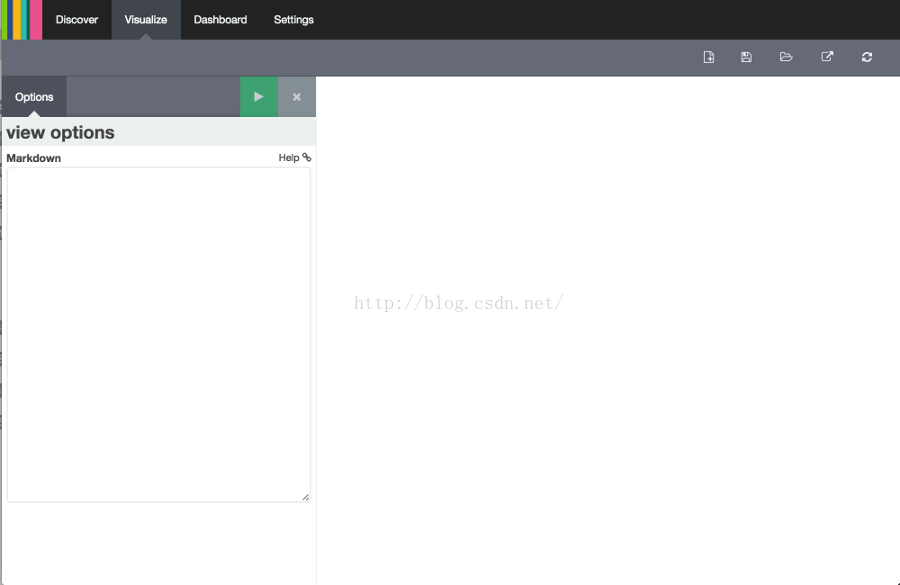


你可以通过点击或者拖拽来导航地图，放大缩小地图用https://img-blog.csdn.net/20160203142409087?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQv/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center按钮，或者点击**Fit Data Bounds**https://img-blog.csdn.net/20160203142527635?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQv/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center按钮来缩小到包含所有点的最低级别。一个绿色的椭圆的过滤器显示在查询框的正下方。



鼠标移到过滤器显示一些控制工具：切换、定位、翻转、删除过滤器。用Map Example名字保存这个图表。

最后我们用一些标记小部件来展示我们的仪表盘。点击**New Visualization**按钮然后点击**Markdown widget**按钮来显示一个非常简单的标记字段输入框：

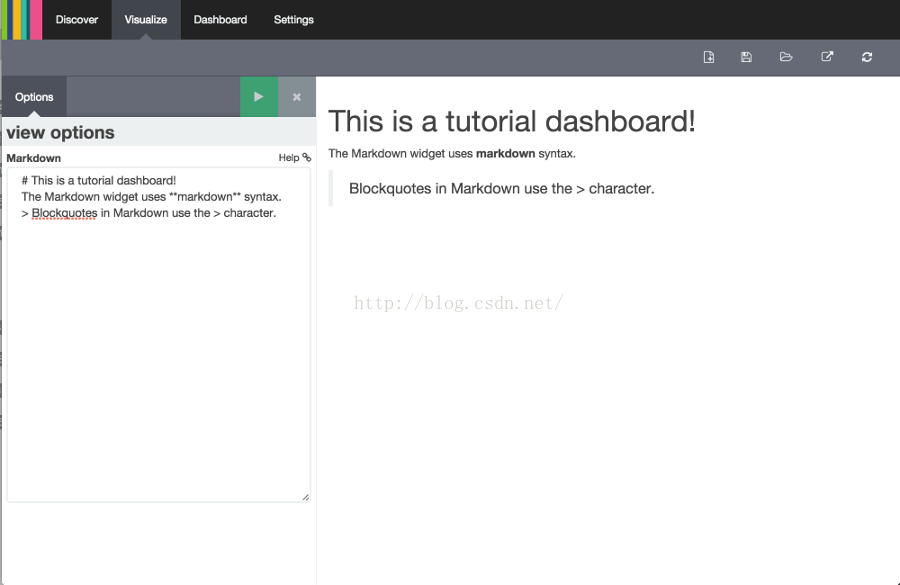


在那个字段输入如下文本：

**[javascript]** [view plain](https://blog.csdn.net/ming_311/article/details/50619859) [copy](https://blog.csdn.net/ming_311/article/details/50619859)

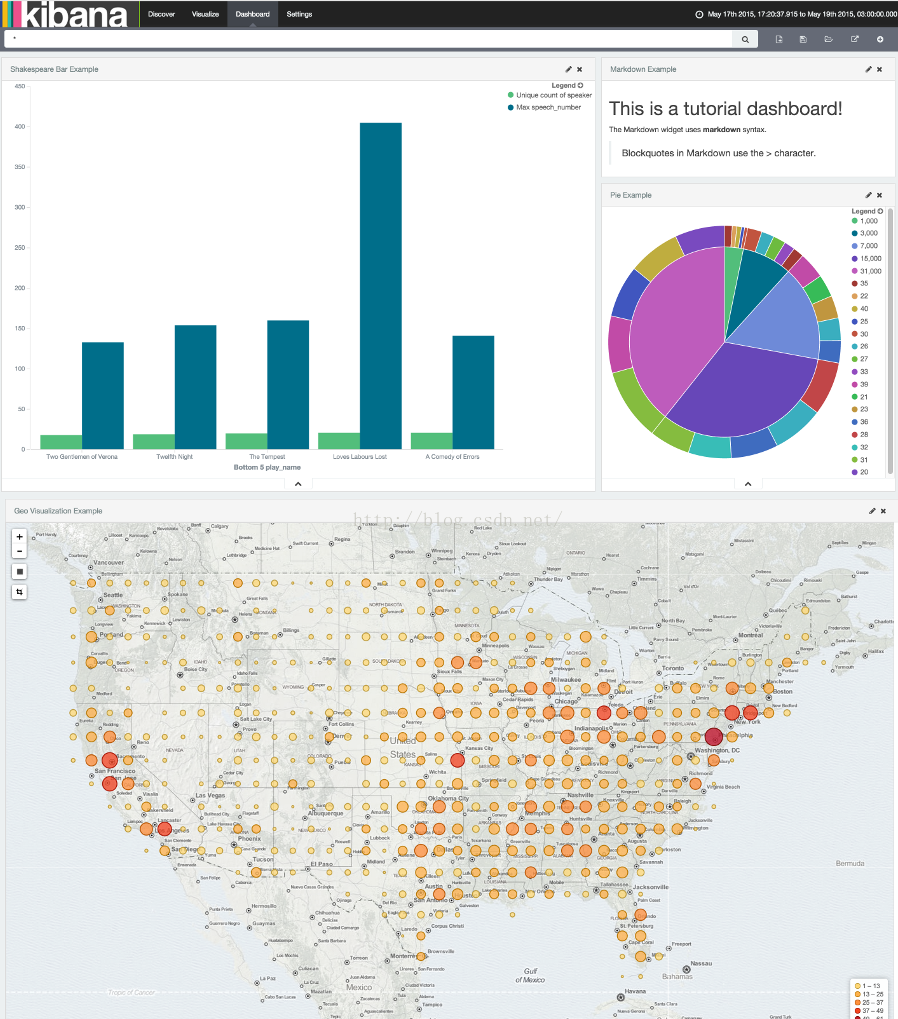
1. # This is a tutorial dashboard!
2. The Markdown widget uses \*\*markdown\*\* syntax.
3. > Blockquotes **in** Markdown use the > character.

点击Apply changes按钮https://img-blog.csdn.net/20160203111227371?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQv/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center在预览面板显示出渲染出来的标记效果。

  
用Markdown Example名称保存这个图表。

**3.全部放入仪表盘**

      一个Kibana仪表盘是许多图表的集合允许你整理和分享。点击Dashboard页面以开始，点击搜索框最右边的**Add Visualization**按钮，显示出已保存图表的列表。选择Markdown Example、Pie Example、Bar Example和Map Example,然后点击底部小小的箭头关闭列表。你可以通过点集合拖拽标题条移动各个图表的容器，通过拖拽图表容器右下角调整容器大小。你的样例仪表盘最终看起来差不多是这样：



        点击**Save Dashboard** 按钮，然后为仪表盘命名为Tutorial Dashboard。你可以通过点击**Share** 按钮来显示HTML嵌入代码或者是一个定向链接分享一个保存的仪表盘。