A description of the design and development of a histogram chart, which is a statistical process control chart. Report authors can use the steps in this article as a template for designing their customized histogram.

View more content in this series

Introduction

Purpose of Document

This document describes the design and development of histogram chart which is a statistical process control chart. This article contains all the steps to design a dynamic histogram that report authors can use as a template for designing their customized histogram.

Applicability

This document applies to IBM Cognos BI 10.2 and higher.

Assumptions

Target readers should be an IBM Cognos report author and have good knowledge on IBM Cognos Report Studio 10.2.1.1. In addition, this document makes use of the IBM Cognos BI samples, in particular the GO Data Warehouse (analysis) package.
Creating a Histogram

A histogram is a type of graph that has wide application in statistics. Histograms allow a visual interpretation of numerical data by indicating the number of data points that lie within a range of values called a **class** or a **bin**. The frequency of the data that falls in each bin is depicted by the use of a bar.

In this example, we used the **Returned Items** namespace under the **Sales and marketing (analysis)** folder of **GO Data Warehouse (analysis)** package. Return quantity fact against calendar date is used for the bin creation. The remainder of this document lists the steps to be used for designing the dynamic histogram report.

1. Open **IBM Cognos Report Studio** and create a new **Column Chart** report using the **GO Data Warehouse (analysis)** package.
2. In the **Query Explorer** select the chart query **Query 1**. Drag **Date** from the following location,
   
   `[go_data_warehouse].[Sales and Marketing (analysis)].[Returned items].[Time].
   [Time].[Day].[Date]`

3. Add **Return quantity** from the following location,
   
   `[go_data_warehouse].[Sales and Marketing (analysis)].[Returned items].[Returned items].
   [Return quantity]`
Figure 1 - Returned items namespace showing the Return quantity and Date query items

4. From Toolbox tab, add a Data Item to Query1 for each of the 19 items listed in Table 1.
<table>
<thead>
<tr>
<th>Data Item Name</th>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Date</td>
<td>maximum([Date] for report)</td>
<td>Fetch maximum date</td>
</tr>
<tr>
<td>Min Date</td>
<td>minimum([Date] for report)</td>
<td>Fetch minimum date</td>
</tr>
<tr>
<td>Mean</td>
<td>average([Return quantity] for report)</td>
<td></td>
</tr>
<tr>
<td>Median</td>
<td>median ([Return quantity] for report)</td>
<td></td>
</tr>
<tr>
<td>Count of obs</td>
<td>Count ([Return quantity] for report)</td>
<td>Count of observations</td>
</tr>
<tr>
<td>Min_report</td>
<td>minimum ([Return quantity] for report)</td>
<td></td>
</tr>
<tr>
<td>Max_report</td>
<td>maximum ([Return quantity] for report)</td>
<td></td>
</tr>
<tr>
<td>Steps</td>
<td>if (?No of bins? =0) then (([Range])/[No. of Bin Intervals_Max 10]) else if (?No of bins? &gt;0 and ?Steps?=0) then ([Range])/?No of bins? else if (?No of bins? &gt;0 and ?Steps?&gt;0 and ?Bin_Radio_Button?=1) then ([Range])/?No of bins? else (?Steps?)</td>
<td>Steps is difference between the bin ranges.</td>
</tr>
<tr>
<td>Default Steps</td>
<td>[Range]/[Default_No of Bin Intervals]</td>
<td></td>
</tr>
<tr>
<td>Bin Interval</td>
<td>(CEILING(([Return quantity]-[Min_report])/[Steps]))</td>
<td></td>
</tr>
<tr>
<td>Bin Numbers</td>
<td>if ([Bin Interval] = 0) then (1) else ([Bin Interval])</td>
<td></td>
</tr>
<tr>
<td>Frequency</td>
<td>count([Bin Numbers] for [Bin Numbers])</td>
<td></td>
</tr>
<tr>
<td>Range</td>
<td>[Max_report]-[Min_report]</td>
<td></td>
</tr>
<tr>
<td>SD</td>
<td>standard-deviation-pop ([Return quantity] for report)</td>
<td>Standard Deviation</td>
</tr>
<tr>
<td>Default_No of Bin Intervals</td>
<td>ceiling (cast ((ceiling (5*([Range]/[SD]))/10), integer)*10)</td>
<td></td>
</tr>
<tr>
<td>No. of Bin Intervals</td>
<td>if (?Steps?=0) then (cast (ceiling (5*([Range]/[SD]))/10), integer)*10) else (ceiling ([Range]/?Steps?))</td>
<td></td>
</tr>
<tr>
<td>No. of Bin Intervals_Max 10</td>
<td>if (?No of bins?&gt;0 and ?Steps? =0) then (?No of bins? ) else if(?No of bins?&gt;0 and ?Steps?&gt;0 and ?Bin_Radio_Button?=1) then (?No of bins?) else ((No. of Bin Intervals))</td>
<td></td>
</tr>
<tr>
<td>Mean_bin no</td>
<td>(CEILING([Mean]-[Min_report])/[Steps])</td>
<td>This is to find out the bin number or the bar which contains the Mean.</td>
</tr>
</tbody>
</table>
5. In **Query 1**, create the following three detail filters. The filter names are **date**, **Param_Min** and **Param_Max** respectively.
   
a. **[Returned items].[Time].[Time].[Day].[Date] in_range ?date?**
   
   An alternative to using `in_range` is to use two separate date prompts and the `between` operator.

b. **[Return quantity] >= ?Param_Min?**

   This filter is to allow users to select their own preferred minimum range from which they want to plot the histogram.

c. **[Return quantity] <= ?Param_Max?**

   This filter is to allow users to select their own preferred maximum range till which they want to plot the histogram.

6. In **Query Explorer**, rename **Query1** to **Base_qry**.

7. From the **Toolbox** pane, drag a **Query** object under **Base_qry**.

8. Rename the new query as **Qry_Dynamic BIN creation**.

9. Click and drag **Base_qry** to the right side of **Qry_Dynamic BIN creation** in order to create a query reference. The result should appear as shown in Figure 2.

   **Figure 2 – Query Explorer showing the Qry_Dynamic BIN creation query referring to the Base_qry query**

10. Double click on **Qry_Dynamic BIN creation** query. Drag the following from **Base_qry** to the **Data Items** section.

    - Bin Numbers
    - Min_report
    - Max_report
    - Range
    - SD
    - Steps
    - Default No. of Bin Intervals
    - No. of Bin Intervals_Max 10
    - Mean_bin no
11. Using the Qry_Dynamic BIN creation query, create a new Data Item for each of the four items listed in Table 2 in addition to the data items created in the previous step.

<table>
<thead>
<tr>
<th>Data Item Name</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running-Count</td>
<td>running-count ([Default No. of Bin Intervals_Max 10])</td>
</tr>
<tr>
<td>Range_From (All)</td>
<td>(((Running-Count)-1)*[Base_qry].[Steps])+ [Base_qry].[Min_report]</td>
</tr>
<tr>
<td>Range_To (All)</td>
<td>((Running-Count)*[Base_qry].[Steps])+[Base_qry].[Min_report]</td>
</tr>
<tr>
<td>Frequency</td>
<td>count((CEILING(([Base_qry].[Return quantity]-[Base_qry].[Min_report])/[Base_qry].[Steps])) for [Base_qry].[Bin Numbers])</td>
</tr>
</tbody>
</table>

12. From Query Explorer, drag a Query object from the Toolbox pane under Qry_Dynamic BIN creation.
13. Rename the new query as Bin Intervals_Obs.
14. Click and drag Qry_Dynamic BIN creation to the right side of Bin Intervals_Obs to create a query reference. The result should look similar to Figure 3.

Figure 3 – Query explorer showing Bin Intervals_Obs query referring to the Qry_Dynamic BIN creation query

15. Double click on Bin Intervals_Obs query and drag the following from Qry_Dynamic BIN creation to the Data Items section.
   - Bin Numbers
   - Frequency
   - Mean_bin no

16. From Query Explorer, drag a Query object from the Toolbox pane under Bin Intervals_Obs.
17. Rename the new query as Bin Intervals_All.
18. Click and drag Qry_Dynamic BIN creation to the right side of Bin Intervals_All to create a query reference. The result should look similar to Figure 3.
19. Double click on **Bin Intervals_All** query and drag the following from to the **Data Items** section.
   - Running-Count
   - No. of Bin Intervals_Max 10
   - Frequency
   - Steps
   - Min_report
   - Range_From(All)
   - Range_To (All)
   - Count of obs

20. From Query Explorer, drag a **Join** object from the **Toolbox**.

21. Rename **Query1** to **Qry_Final**.

22. Drag the **Bin Intervals_Obs** query to the top dotted box and drag the **Bin Intervals_All** query to the lower dotted box. Result should appear as shown in Figure 5.
Figure 5 – Query explorer showing the Bin Intervals_All and Bin intervals_Obs queries joined to create the Qry_Final query

23. Double click on **Qry_Final** and drag the following data items from the **Bin Intervals_Obs** and **Bin Interval_All** queries to it. In addition, set the properties for each data item.

- **Running-Count**
  - Aggregate Function: Not Applicable
  - Rollup Aggregate Function: None
- **Range_From (All)**
  - Aggregate Function: Not Applicable
  - Rollup Aggregate Function: None
- **Range_To (All)**
  - Aggregate Function: Not Applicable
  - Rollup Aggregate Function: None
- **Mean_bin no**
  - Aggregate Function: Not Applicable
  - Rollup Aggregate Function: Automatic
- **Count of obs**
  - Aggregate Function: None
  - Rollup Aggregate Function: None

24. Create the five **Data Items** listed in Table 3 in **Qry_Final**. For all the data items except the one named Frequency, set the **Aggregate Function** property to Not Applicable and the **Rollup Aggregate Function** to None.
<table>
<thead>
<tr>
<th>Data Item Name</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>coalesce ((total([Bin Intervals_Obs].[Frequency] for [Bin Intervals_Obs].[Bin Numbers])) ,0)</td>
</tr>
<tr>
<td>Range_From (All)_Varchar</td>
<td>cast(cast([Range_From (All)], decimal(10,2)),varchar(15))</td>
</tr>
<tr>
<td>Range_To (All)_Varchar</td>
<td>cast(cast([Range_To (All)], decimal(10,2)),varchar(15))</td>
</tr>
<tr>
<td>Range Name (All)</td>
<td>[Range_From (All)_Varchar]</td>
</tr>
<tr>
<td>Bin Containing Mean Value</td>
<td>if ([Mean_bin no]=[Bin Interval_All].[Running-Count]) then ([Bin Intervals_Obs].[Frequency]) else (0)</td>
</tr>
</tbody>
</table>

25. Add the following detail filters to Qry_Final.
   - [Bin Interval_All].[Running-Count]<=[Bin Interval_All].[Default No. of Bin Intervals_Max 10]
   - [Bin Interval_All].[Count of obs]>=?No of bins?
   - [Bin Interval_All].[Default No. of Bin Intervals_Max 10]<=[Bin Interval_All].[Count of obs]

26. Click on the Page Explorer bar and then select the report page Page1.

27. Create a date range prompt by dragging the Date Prompt tool from Toolbox to the report page. Select the Use Existing Parameter option and choose the parameter name date which we created in back in Step 5. Click Finish.

28. Parameter should be an optional parameter.

29. In the Properties pane for the date prompt, under the General section set Select UI to Edit box, Calendar Type to Gregorian and Range to Yes.
30. Create a value prompt by dragging the **Value Prompt** tool from the **Toolbox** below the date prompt that was just added.

31. Provide the parameter name **Bin_Radio_Button**. In the **Properties** pane under the **General** category, set **Required** to **Yes**, **Auto-Submit** to **Yes** and **Select UI** as **Radio button group**.

---

**Figure 6** – Properties pane showing date prompt General properties, Required is set to No, Select UI is Edit box, Calendar Type is Gregorian and Range is set to Yes
32. Under the **Data** section of the **Properties** pane, set the **Static Choices** property for the newly created value prompt. The static choices have **Use** values of 1 and 4 and **Display** values of **Number of bins** and **Bin Interval** respectively.

33. Under the **General** section of the **Properties** pane, set the **Default Selections** property to 1 which represents the default value as **Number of bins** in the value prompt.
34. Drag a **3x3 table** from the **Toolbox** beside the newly created value prompt. In the first column of the second row, drag a **Text Item** and type **Number of bins** as the value. Similarly in first column of the last row, drag a **Text Item** and type **Bin Interval** as the value. Use Text Items in first row, second and third columns to show the text **Default Value** and **User Selected Value** respectively.

35. Create a **Value Prompt** for the number of bins in last column of the second row of the table. When creating the parameter, select the **Use existing parameter** option, chose parameter name **No. of bins** and click **Finish**.

36. Select the value prompt and under the **Properties** pane in the **General** section, set **Required** to **Yes**, **Multi-Select** to **No** and **Auto-Submit** to **No**.

   **Figure 9 – Properties pane for the value prompt showing Required set to Yes, Multi-Select to No and Auto-Submit to No**

37. Set the **Static Choices** property with **Use** values of **0, 10, 15, ....50** - each value between 10 and 50 having a difference of 5. Set the **Display** values to be the same as the **Use** value except for the Use value of 0 which has the Display value of **Select No. of Bin Interval**.

   In this example the number of bins is limited to 50 but can be modified higher or lower as required.
Figure 10 – Static Choices window showing Use and Display values to select the number of bins

38. For the **Default Selections** property under the **General** section, specify a value of 0 in order to show the text **Select No. of Bin Interval**.
39. Create a **Text Box Prompt** in last column of the third row of the table. Select the **Use Existing Parameter** option and choose the parameter name **Steps** from the drop down.
40. Select the text box prompt and under the **Properties** pane in the **General** section set **Required** to **Yes**, **Multi-Select** to **No** and **Default Selections** to **0**.

Figure 11 – Properties pane for the text box prompt showing Required set to Yes, Multi-Select to No and a defined default selection

41. Create a **2x2 table** below the previous table for user to enter minimum and maximum values for the data set for which they want to plot the histogram. In the first column of the second first, drag a **Text Item** and type **Minimum** as the value. Similarly in first column of the second row, drag a **Text Item** and type **Maximum** as the value.
42. Create two **Text Box Prompts** into the table cells beside the text items just created. Select the **Use Existing Parameter** option for both the prompts and choose the parameter names
Param_Min and Param_Max (created earlier in Step 5) from the drop down for minimum and maximum respectively. Click Finish.

Figure 12 – Text box prompts placed into the 2x2 table

43. Create singletons using the Base_qry data items Default_No of Bin Intervals and Default_Steps. In the 3x3 table created earlier, place the Default_No of Bin Intervals singleton in row two, column two and the Default_Steps singleton below it in row three, column two. This is illustrated in Figure 13.

Figure 13 – The Default_No of Bin Intervals and Default_Steps singletons placed into the 3x3 table

44. To design the descriptive statistics table, use a 2x10 table. In the first column add a Text Item into each cell and set the cell value with following values.
   • Descriptive Statistics Table
   • Count of Observations
   • Mean
   • Median
   • Minimum
   • Maximum
   • Range
   • Standard Deviation
   • Start Date
   • End Date
45. In the second column use a **Text Item** in the first row and give it a value of **Values**. In subsequent cells create **singletons** using **Base_qry** with following Data Items as shown in Figure 14.
   - Count of obs
   - Mean
   - Median
   - Min_report
   - Max_report
   - Range
   - SD
   - Min Date
   - Max Date

**Figure 14 – The descriptive statistics table with the singletons**

46. Select each singleton and set the **Properties** property by selecting all the check boxes.

**Figure 15 – The Properties property for a singleton with all data items checked**

47. Change the query of the Column Chart to **Qry_Final** and in the **Categories (x-axis)** drag in the **Range Name (All)** data item and in the **Series (primary axis)** drag the **Frequency** and **Bin Containing Mean Value** data items.
Figure 16 – Properties pane showing Query of the chart is set to Qry_Final and the x-axis and primary axis are filled with data items

48. Set the Trendlines property by selecting the chart and from the Properties pane under the Chart Annotations section, click on Trendlines.

49. In the Trendlines window create a new polynomial trendline and fix the order to 6 as shown below in Figure 17.

Figure 17 – Properties pane of chart showing trendline of type polynomial and order 6

50. Run the report and it should appear similar to the one shown in Figure 18. As with any Report Studio report, additional formatting such as applying color and fonts can be applied as required.
Figure 18 – Report being rendered in IBM Cognos Viewer
About the authors

Kalyani Raut

Kalyani is a Cognos BI developer in the Predictive Maintenance and Quality team (BA). She has worked on Cognos FM modeling and designed various types of Report Studio reports. She has 8 years of experience.

Swamy S T

Swamy is a Cognos BI developer in the Predictive Maintenance and Quality team (BA). He has worked on the creation of various types of reports and various entities in Cognos and BI tools. He has 6 years of experience.

Mohit Sewak

Mohit Sewak is a Predictive and Advanced Analytics Solution Architect and SME at IBM, India Software Labs. He holds several patents and publications in his field of expertise and has rich experience helping some of the most valuable clients worldwide with using advanced analytics and BI solutions to make better business strategy, improve operations, and optimally serve their customers. In his current role he is involved in envisioning and implementation of advanced analytics capability into IBM's Predictive Maintenance and Quality offering. Mohit is academically an MBA and is pursuing MS. He holds several professional certifications like Certified Supply Chain Professional (CSCP), Certified Project Management Professional (PMP), and certified into methodologies and tools like Lean Six Sigma, SPSS, SAS, R, Hadoop, etc.

© Copyright IBM Corporation 2014
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)