Integrating Oracle Advanced Analytics into OBIEE Dashboards

You will have two VMs assigned to you for this hands-on training lab, a windows-based client VM and a server VM. The client VM will contain the majority of the software which we will use in the lab including RStudio, SQL Developer, and OBIEE’s BI Admin tool. You will open connections to your own individual server VM through those programs and through the ODBC system administration tool.

To open a connection to your client VM, use Remote Desktop and enter the IP address of your Client VM.

User name = Administrator
Password = odtug2015
Click OK.

You may see a warning. Ignore it and click Yes.
In this exercise, we will create an analysis in R including a table and a custom box plot visualization and then set up an automatic work flow in Oracle database to spin up an R instance on the database server machine and publish the table and graph on an OBIEE dashboard.

We will start with the original R script. We will use the RStudio development environment to establish a connection to our database server VM and then run our script.

Launch RStudio by double-clicking the RStudio icon on your desktop.
You should see an R script in the upper left hand window pane as shown in the screen shot below. Highlight the first line of the script and click the “Run”. The library(ORE) function will load the ORE libraries that have already been installed on your client VM. This enables you to run custom functions that translate R code into SQL code through the Oracle R Enterprise transparency layer.

After running library(ORE) your console panel should look like the below screen.
Highlight the second line of the script `library(ROracle)` and the third line of the code `Sys.setenv(TZ = "GMT")` and click “Run”. You can either execute R scripts one line at a time or highlight several lines and execute them together. Your console should look like the screen shot below.

Next we will connect to your database server VM. Highlight the IP address in the connection string using the `ore.connect` function as shown below and paste in the IP address for your server VM. Click the “Run” button to run the `ore.connect` line of code and connect to your server VM.

```r
library(ORE)
library(ROracle)
Sys.setenv(TZ = "GMT")
ore.connect(user = "ruser", service_name = "pcborcl", host = "52.0.149.17", password = "ruser", all=TRUE)
ore.is.connected()
```
It may take a few minutes for your client VM to connect to your server VM, so please be patient. Once the connection is established, you should see the below warning message in your console (don’t worry about it.)

```
> system.time(
>   ong.connect(user = "rquser", service_name = "pdborc1", host = "52.8.149.87", password = "rquser", all=true)
> )
```

Warning message:
Table RUSER."AIRPORTSCOUNT" contains unsupported data types

Run the `ore.is.connected()` code to confirm that your connection is live and then the `ore.ls()` code to see what objects exist in the rquser schema in the database.

```
> ore.is.connected()
[1] TRUE
> ore.ls()
[1] "INSUR_CUST_LTV_SAMPLE" "MOV_AV"  "NCLUS"  "ONTIME_S"
[5] "ORCL_STOCK"  "TL"
```

Next we will run a script that finds the top 36 busiest airports in the country and creates a box plot graph showing their distribution of their delays for the year 2007. Take a minute to read through the script. Even though you may not be an R expert, you should be able to follow the script. Highlight the entire script and click “Run”.

![Box plot graph](image)

Your screen should look something like the above screen shot. Drag the borders of the plot window to make it larger and to make the graph bigger. We will register our script in the Rscript repository in the database so that Oracle database can automatically spin up an R engine in its database server memory, pull the data from the ONTIME_S table, and create the box plot graph image.

Exit RStudio. (Do NOT save your workspace. Because we have R running on our local client machine and because it’s connected to a remote database server, it’s generally recommended that you don’t save
your workspace. The database is likely to change next time you connect and you don’t want to load old information from a previous session into R’s memory, so it’s typically best to establish a fresh connection every time.)

Next we will register our script in the database using SQL Developer. Find the SQL Developer icon on your desktop and launch it. You should see an `rquser` and a `System` connection on your connection pane. Right-click the `System` connection and select `Properties`.
Replace the IP address in the **Hostname** field with the IP address for your Database Server VM. Click, **Test** then **Save**, then **Connect**.

Right-click the **rquser** connection in the **Connections** pane and follow the same steps. Replace the **Hostname** field with your database server VM IP, click **Test**, **Save**, and **Connect**.

Right-click the **rquser** connection and select **Open SQL Worksheet**.
Click on the **OBIEE ORE SQL Script.txt** tab or navigate to the .txt file on the desktop. Highlight the entire script (ctrl A), copy it (ctrl C), and then paste it onto the rquser SQL Worksheet (ctrl V).

Highlight the top block of the script and run it. Take a moment to read through the script. Notice that while it is written in R, there are some minor syntax modifications to accommodate SQL’s special character syntax. We now have registered our script under the name “BusiestAirports2” in the R script repository in the database. We could have done this from the RStudio interface as well, but chose to do
it in SQL Developer as many DBAs like to have System schema scripts registered in PL/SQL process (the R Script Repository uses the SYSTEM schema.)

```sql
BEGIN
  sys.rqScriptDrop('BusiestAirports2');
  sys.rqScriptCreate('BusiestAirports2',
    'function|dat|{
      outtime <- dat.
      n <- 36
      aggdata <- aggregate(outtime$DEST, ov = list(outtime$DEST), FUN = length)
      minx <- min(head(sort(aggdata$x), decreasing = TRUE), n)
      busiest_airports <- aggdata$Group.[aggdata$x >= minx, drop = TRUE]
      delay <- ontime$AWEDELAY[ondate$DEST win busiest_airports || ov || ondate$YEAR <= 2007]
      dest <- ondate$DEST[ondate$DEST win busiest_airports || ov || ondate$YEAR <= 2007, drop = TRUE]
      pd <- reorder(dest, delay, FUN = median, na.last = TRUE)
      cc <- boxplot(pd, notch = TRUE, col = "gold", cex = 0.5,
        outline = FALSE, horizontal = TRUE, yaxt = "n",
        main = paste("2007 Flight Delays by Airport, top", ", ", busiest", sep = "",
        xlab = "Delay (minutes)", ylab = "Airport"
      )
      labels <- levels(dest)
      text(par("usr")[1] - 3, 1:length(labels), col = 0, adj = 1,
        labels = labels, cex = TRUE, cex = 0.75)
      topbusiest <-
        data.frame(YIELD_25th = quantile(resstats[1,], 0.25),
        MEDIAN = resstats[3,],
        CONF HIGH = resstats[4,],
        ylab = "Airport")
  }
END;
```

Next we will execute a SQL script to create a table output that we can place on our OBIEE dashboard to accompany our Box Plot graph.

Highlight the next block of code on the rquser SQL worksheet and run it by clicking on the green arrow. Take a moment to read through the script and notice how we are using a combination of cursor select statements, column definitions, and the above R script that we just registered. Understanding the exact syntax of the rqTableEval function is beyond the scope of this hands-on lab, but you should be able to follow what is happening.
Notice that we have all the information that is included in our graph, the airport code, minimum delay, in the interquartile range (Q1 and Q3), and the low and high confidence intervals, and the maximum delay.

Next we’ll create our graph and save it as a PNG BLOB image that can be published on an OBIEE dashboard.

Highlight the next block of code and run it.
SQL Developer now has the ability to preview images. Right-click on the (BLOB) field and select Single Record View.

Then click on the pencil icon and then click the Image box.

You should see our box plot graph!
Next we will establish a connection with the BI Server and modify the repository .rpd file.

Find the Data Sources ODBC connections icon at the bottom of your desktop and open it.
Click **Configure** to open the OBIEE connection.

Next, enter your Server VM IP in the **Server** field and click **Next**.
Enter **Admin123** in the password field and click next.

It may take a few minutes to establish a connection, so please be patient. Click **Finish** and close the configuration interface.

Open the BI Admin tool by double-clicking the red Oracle “O” square labeled **Administration** on the desktop.

Click **File**, then **Open**, then select **Online**. In real world settings, we recommend making all the changes in the **Offline** mode, but in the interests of time and avoiding stopping and restarting services, we’re going to do everything in online mode today. Remember, you have your own individual client and your own individual server with no one else on it. Adding and editing physical sources in the repository is not recommended for most all other situations.
Enter **Admin123** as the password for in the **Repository Password** field and in the **Password** field and click **Open**.

In the **Physical** layer panel, click on the **07 – ORE Sample** data source node to expand it.

We will now add a physical table to hold the table that we created in SQL Developer with the data fields that are used to create our Box Plot graph. This demonstrates how to import structured data from an embedded R process that is run in the database into OBIEE.

Right-click the **07 – ORE Sample** node and select **New Object** and then **Physical Table**.
Click Yes you want to check it out.

Name your table BUSIEST_AIRPORTS_2. Choose Select in the Table Type drop down box.

Click the Use database specific SQL radio button and paste the following code into the Initialization String field as shown.

```sql
select UNIQUECARRIER, MINI, Q1, CONF_LOW, MEDIANI, CONF_HIGH, Q3, MAXI
from table(rqTableEval(cursor(select * from ONTIME_S),
    cursor(select 1 "ore.connect" from dual),
    'select cast("a" as varchar2(10)) UNIQUECARRIER, 1 MINI, 2 Q1,
        3 CONF_LOW, 4 MEDIANI, 5 CONF_HIGH, 6 Q3, 7 MAXI
    from dual',"BusiestAirports2"));
```
Go to the Columns tab and click the Green plus sign to define the columns as shown below.

Go ahead and check in your changes by clicking on the check in changes icon in the menu bar. Do NOT check global consistency.

Drag your BUSIEST_AIRPORTS_2 table from the Physical layer to the 08 – ORE Sample node in the Business Model and Mapping layer twice. Name one BUSIEST_AIRPORTS_2_F and the other BUSIEST_AIRPORTS_2_D.
Highlight both tables in the business model logical layer then right-click and select Business Model Diagram and then Selected Tables Only.
Click on the **Join** tool in the tool bar and then click and drag to join the two tables as shown. Then click **OK** and close the business layer diagram.
Open the BUSIEST_AIRPORTS_2_F table by double-clicking it. Then click on the Keys tab. Type Carrier in the Key Name field and select UNIQUECARRIER in the columns field.

Next select Carrier in the Primary Key dropdown as shown and click OK. Repeat the procedure for the BUSIEST_AIRPORT_2_D table.
Drag the BUSIEST_AIRPORTS_2_F table to the R – ORE SAMPLE subject area node in the presentation layer as shown.

Click the Check in Changes icon on the tool bar and check global consistency. (You should have 0 errors and 11 warnings.)
Open a browser window and type the IP address for your Server VM followed by :7780/analytics as shown:

Log in as Prodney/Admin123.

Click on the New icon in the menu bar and select Analysis.

Select the R – ORE Sample subject area.

You should see your BUSIEST_AIRPORTS_2_F table in the left hand selection pane. Open it and drag all the columns to the Selected Columns pane on the Criteria tab. You Criteria tab should look something like this.
Click on the Results tab and Voila! You should have a table with the structured results of our R scripts. Save your analysis for use later on a dashboard.

Return to the BI Admin tool so that we can bring in the Box Plot graph image. We will a similar path to create a table in the physical layer, join it to a second copy in the business layer, and promote to the presentation layer.

Right-click the ORE Sample node and select New Object and then Physical Table. Click Yes you want to check it out.
Name your new table BUSIEST_AIRPORTS_2_IMAGE. Choose Select in the Table Type field.

Click the Default Initialization String button and enter the SQL code below in the field. Take a minute to read the code and understand what is happening. We are specifying and select only two columns, ID and IMAGE. We are using the rqTableEval function to spin up an R engine as an embedded process on the database server machine and to execute our BusiestAirports2 script. We are also specifying the data type for our second column as PNG which is a graphic type that OBIEE can display.

```sql
select   ID, IMAGE 
from  table(rqTableEval(
    cursor(select * from ONTIME_S),
    cursor(select 1 "ore.connect" from dual),
    'PNG','BusiestAirports2'))
```

Click on the Columns tab. We will now create the ID and IMAGE columns for our table. Click on the green plus sign and enter ID in the name field and select INT in the Type drop down field. Repeat the process and create a column named IMAGE of Type LONGVARBINARY and Length 128000.
Go to the **Keys** tab and type **ID** in the **Key Name** field and then select **ID** in the column drop down and click **OK**. Check in your changes and do not check global consistency. You should now see a key icon next to the ID column.

Drag **BUSIEST_AIRPORTS_2_IMAGE** table to the **08 – ORE Sample** node in the business layer twice. Join the **BUSIEST_AIRPORTS_2_IMAGE** and **BUSIEST_AIRPORTS_2_IMAGE#1** tables in the Business Layer diagram using the join tool as we did in the previous example.

Expand **BUSIEST_AIRPORTS_2_IMAGE** and double-click the logical table source icon in the **Sources** folder. On the **Column Mapping** tab, click on the **IMAGE** column and then on the **Edit Expression** icon.

Place your cursor at the very beginning before the “07-ORE…” string and type `lookup()`.

Place a comma at the end of the “...IMAGE” string. Then double-click **ID** in the **Columns** field on the left to add it to the expression. Close the expression with a “)” parenthesis mark at the very end. Your screen should look like this. Click **OK** and **OK**.
Double-click on the IMAGE column.

Click on the **General** tab. Click on **Set** and the ID column for both the **Sort Order Column** and **Descriptor ID Column**.

Drag **BUSIEST_AIRPORTS_2_IMAGE** table to **R_ORE Sample** node in the **Presentation** layer.

Check in your changes and check Global Consistency. You should have 0 Errors and 11 Warnings.

Save your changes and exit BI Admin tool.
Open your browser and sign back in to OBIEE Prodney/Admin123 if you exited. Click the Administration link in the menu bar and click the Reload Files and Metadata link.

Click on New then Analysis and then choose the ORE Sample subject area.
Open the **BUSIEST_AIRPORTS_2_IMAGE** folder and add both columns to the Selected Columns pane on the Criteria tab.

Click on the **Results** tab.

You should see your box plot graph! Save your analysis.

Click on Catalog in the menu bar and then edit page 1 of My Dashboard.
Navigate to your two saved analyses in the catalog directory on the left hand side and drag them on to your dashboard.

Click Save and then Run.
We’re Finished!