Bluemix Hands-On Exercise

Analytics Using R and R Studio in Bluemix with dashDB Analytics Warehouse

Version : 4.00
Last modification date : 30 September 2014
Owner : IBM Ecosystem Development

Work with analytics R scripts and R Studio for a scenario to analyze telecom industry customer churn based on existing customer data. Once you debug and test the script in R Studio you can deploy the scenario to a production Bluemix analytics warehouse environment in dashDB, including customer data in data warehouse tables and your deployed R analytics script modified to use the data from the warehouse.

Prerequisites:

- Register on IBM Bluemix at http://bluemix.net

Hands-on Exercise: Big Data & Analytics – Analytics Using R and R Studio in Bluemix with dashDB Analytics Warehouse

Get started using R scripts for an analytics business industry scenario for Telecom, with an example applying analytics to customer data to identify customer churn likelihood. Use R Studio to work with the R analytics script, step thru running and testing the script, and the generated analytics reports and plots. Then, deploy the analytics script and associated customer data to the dashDB data warehouse, make changes needed to access the data from the warehouse and test. We include references to resources to get started and go further using R analytics and the dashDB data warehouse service as well.

Part 1 – Work with the R Script in R Studio to perform statistical analysis for customer churn from the provided customer data

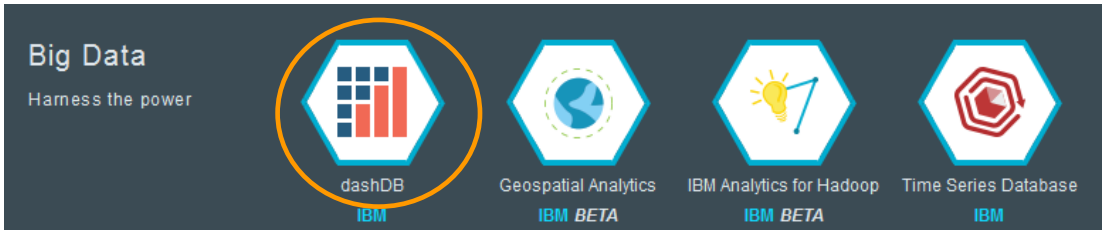
Step 1 Add the dashDB analytics data warehouse to an existing application in Bluemix:

If you do not have any applications available, first create one from one of the provided runtimes in the Bluemix catalog, for instance Node.js:

Add the **Analytics Warehouse** service to your application from the Bluemix catalog:

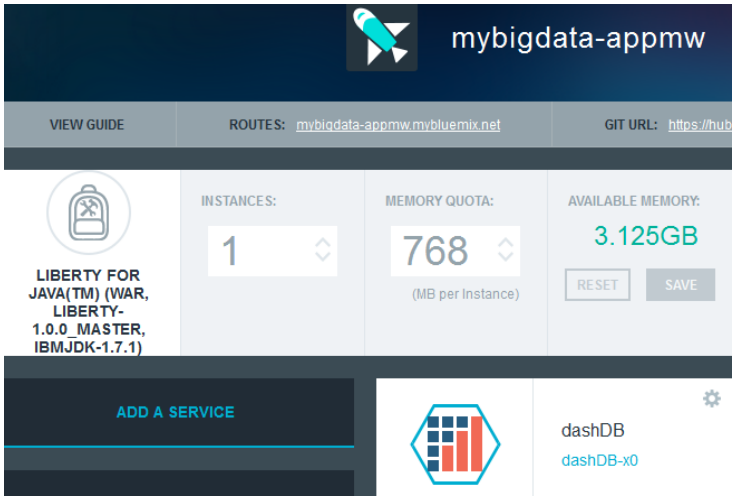
In the Application Overview for your application, click on the **ADD A SERVICE** link. The SaaS Services in the Bluemix catalog are listed.

In the **Big Data** section, locate the **dashDB** warehouse service and select it.

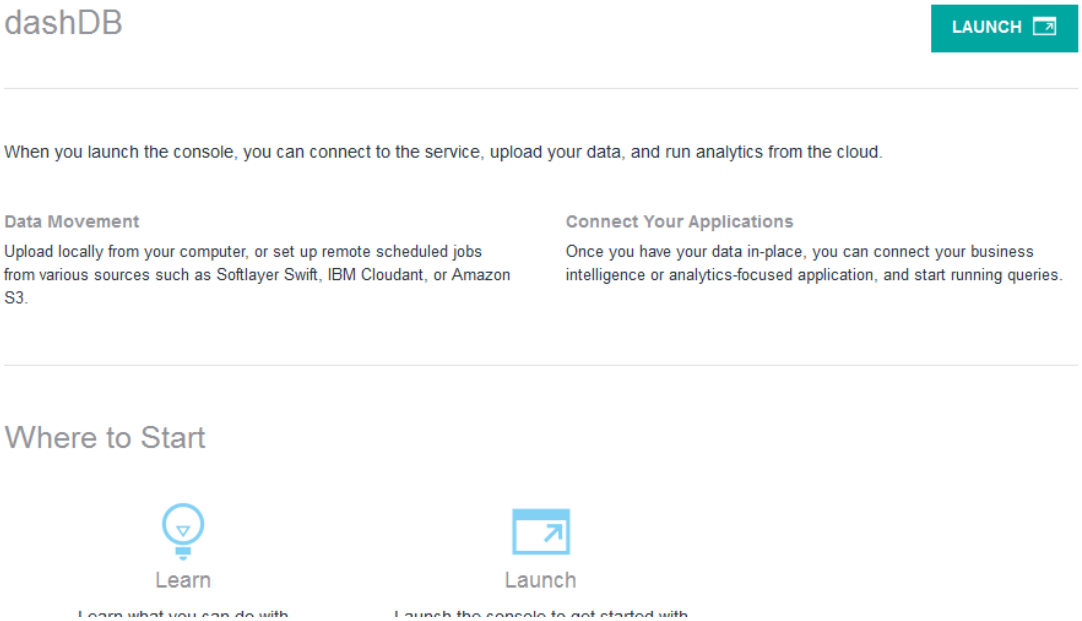


Step 2 In the "Add service" dialog, select the application you would like to add the services to and also provide a name for the instance of the service and click **Create.** Note that Bluemix will provide a default name for the instance of the service, or you can assign a specific name you choose.

1. If you are prompted to restart the application select **OK.** The service instance for **dashDB Warehouse** is now added to the application

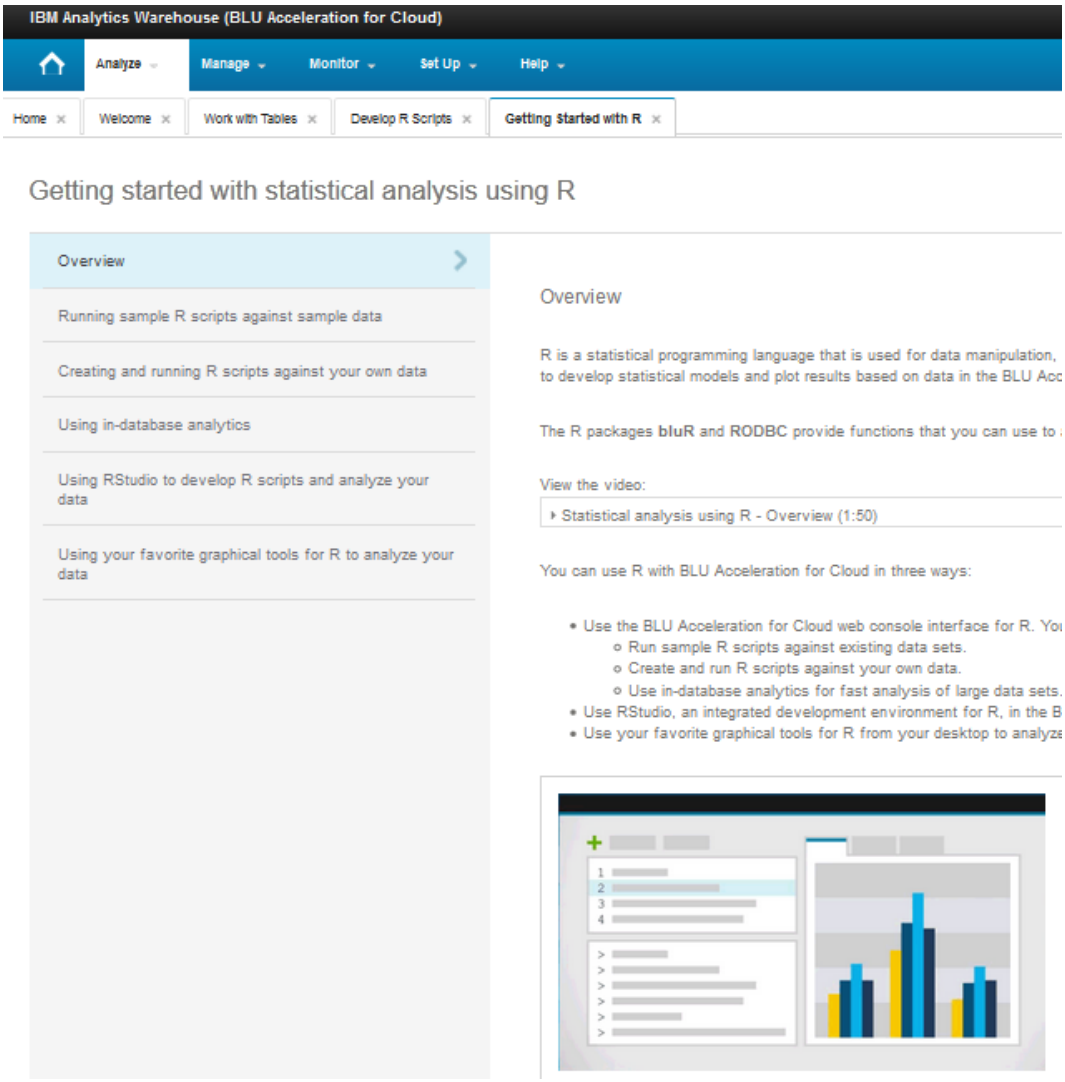


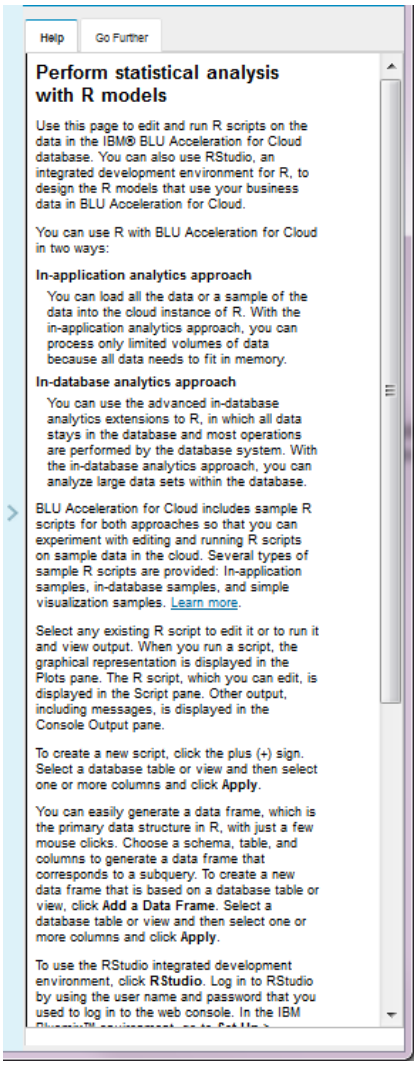
2. Select the **dashDB** warehouse service instance in your application to access the console for the service where you can connect analytics based applications and load/manage data in the warehouse.



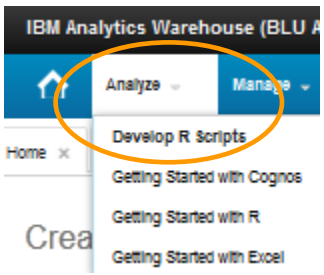
3. Click **Launch** to start the dashDB warehouse console

Note that you can work with data in dashDB warehouse tables, load data from external sources, design databases and perform business and statistical analysis using Cognos, R and Excel.



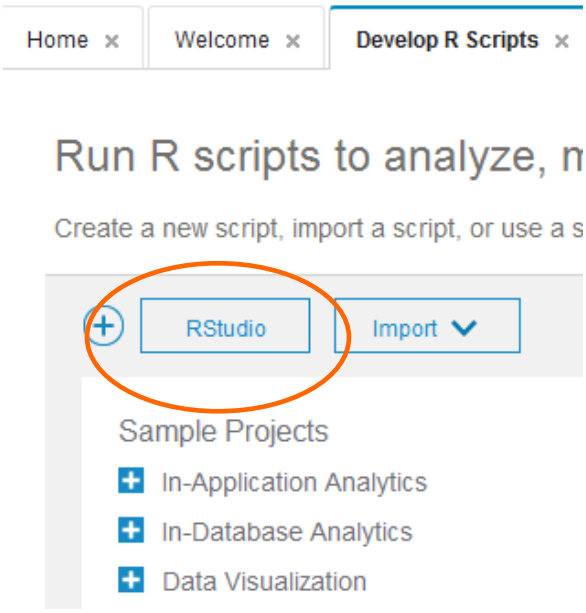


4. Lets go to the area in the console for developing analytics scripts and reports – in the console, under **Analytics**, select **Develop R Scripts**

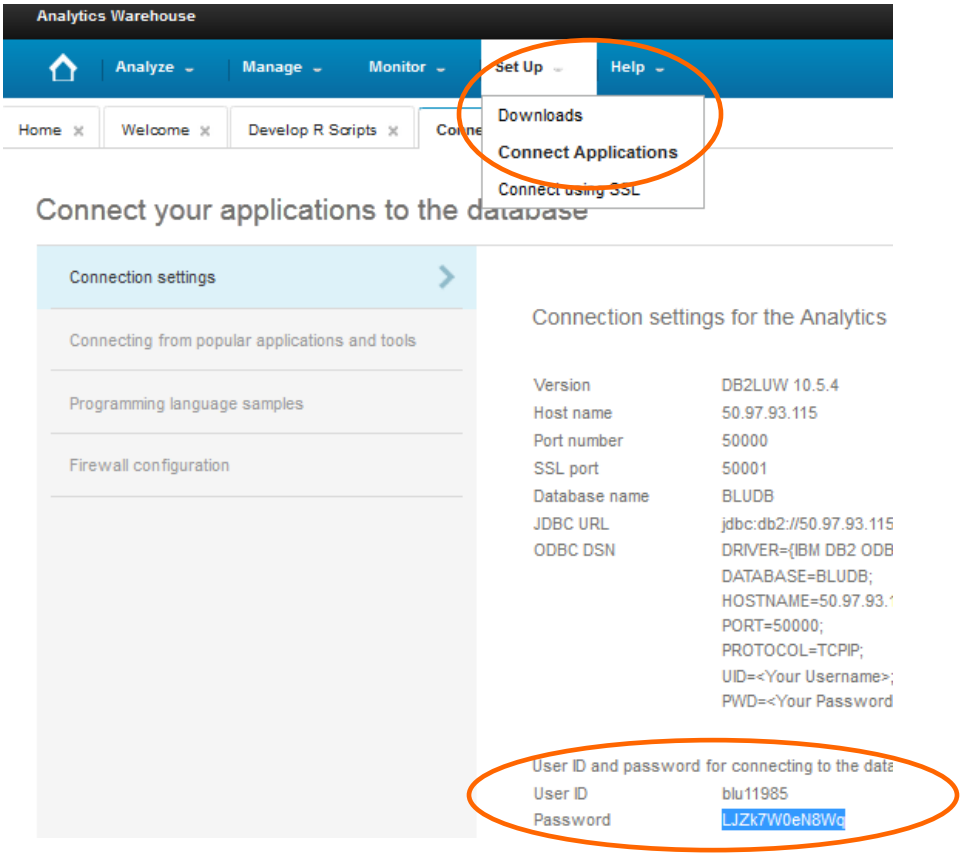


5. The Analytics Warehouse includes an integrated instance of RStudio running in the cloud you can use to develop, test and debug your analytics R scripts and related projects, plots and graphs. We'll access the project in R Studio for the sample project for Customer Churn:

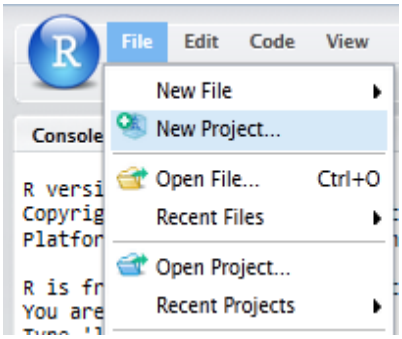
   In the Analytics Warehouse console tab for **Develop R Scripts**, at the top left, select **R Studio**:



6. When the login for R Studio appears, copy/paste the userid and password from Warehouse console – you can view connection information in the **Set up > Connect Applications** option in the Warehouse console:



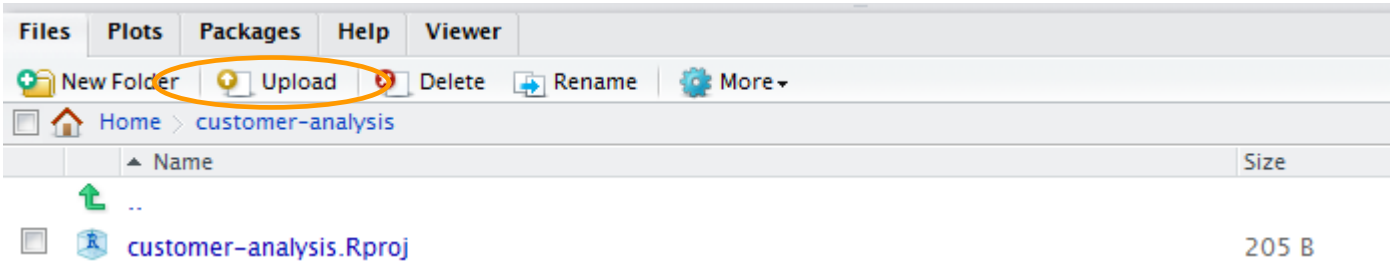
7. Create a new project – select **File > New Project**



Select **New Directory > Empty Project** and enter a directory name for the project, for instance **customer-analysis.** You can leave the prompt for "**Create project as a subdirectory of"** as-is, since it will be created under the root directory in R Studio. Click **Create project**

8. We will now import the R script for the customer churn analysis case – it is available here. If the link does not become active, the download URL location is:
https://www.ibm.com/developerworks/community/groups/service/html/communityview?communityUuid=18b8491b-bc76-469d-806d-e6982265ace7#fullpageWidgetId=W273fd64f1ecb_4104_a0aa_8946ed9765b0&file=943a183a-8645-4c02-a369-2c4b49662988
Download the file to your local filesystem.

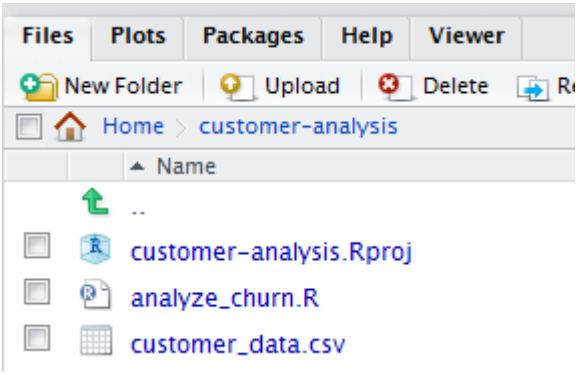
9. In the browser tab for R Studio, in the File Explorer in your workspace area at the lower right, click **Upload**



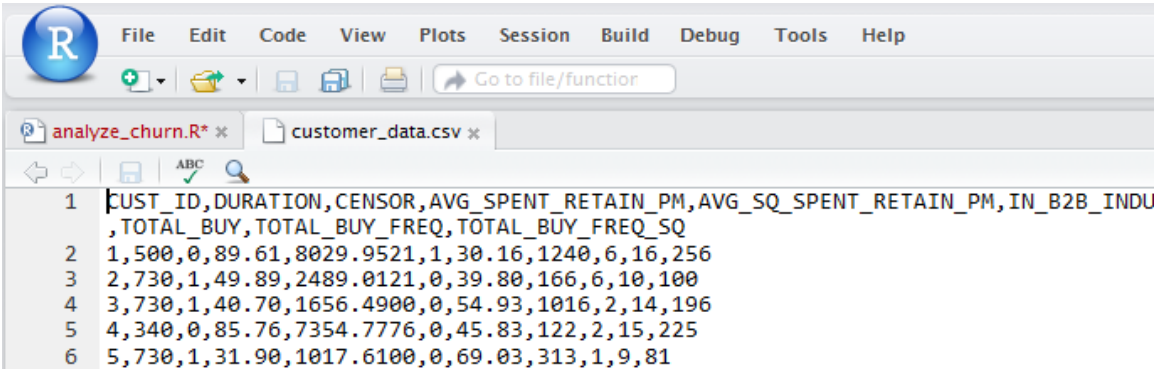
10. Click **Browse,** navigate to the location where you downloaded the source code file for the R script, select it and click **OK**

11. Repeat the two previous steps to import the customer data CSV file which you can access here. If the link does not become active, the download link URL is:
https://www.ibm.com/developerworks/community/groups/service/html/communityview?communityUuid=18b8491b-bc76-469d-806d-e6982265ace7#fullpageWidgetId=W273fd64f1ecb_4104_a0aa_8946ed9765b0&file=9aabb2f3-4d41-472f-b392-4b088394ad67

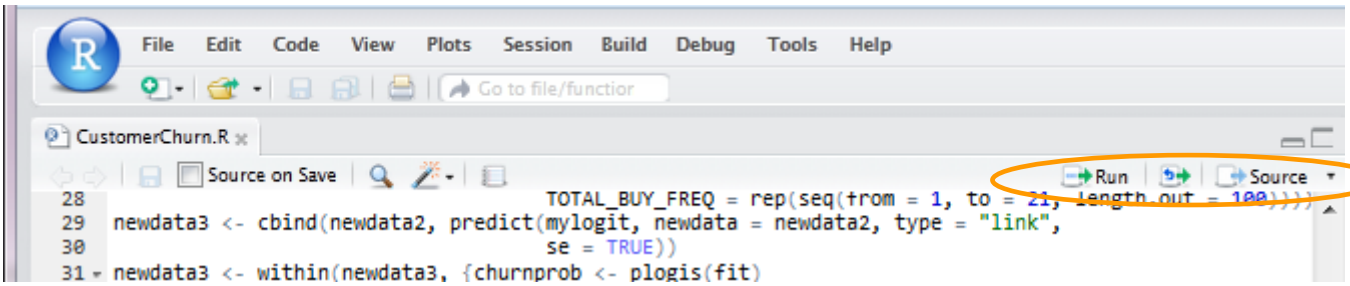
You should now have both the R script and the customer data CSV file in the project in your workspace



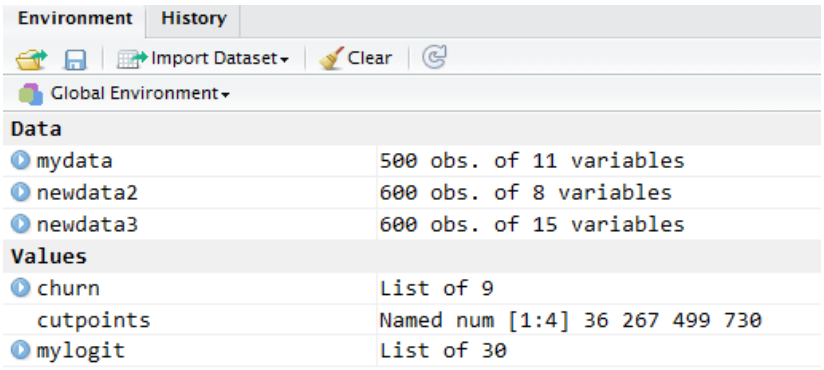
12. Examine the customer data – select the CSV file to open it in the editor window, it will have attributes for customer telecom usage activity summary data such as call durations, costs, B2B customer type, etc……



13. To run the script in R Studio, click on the script, you can start working the script code, debug, test and use other integrated R Studio tools with your scripts. We'll single-step thru the script: Click **Run.** In single-step test run mode, you'll need to click **Run** for each step



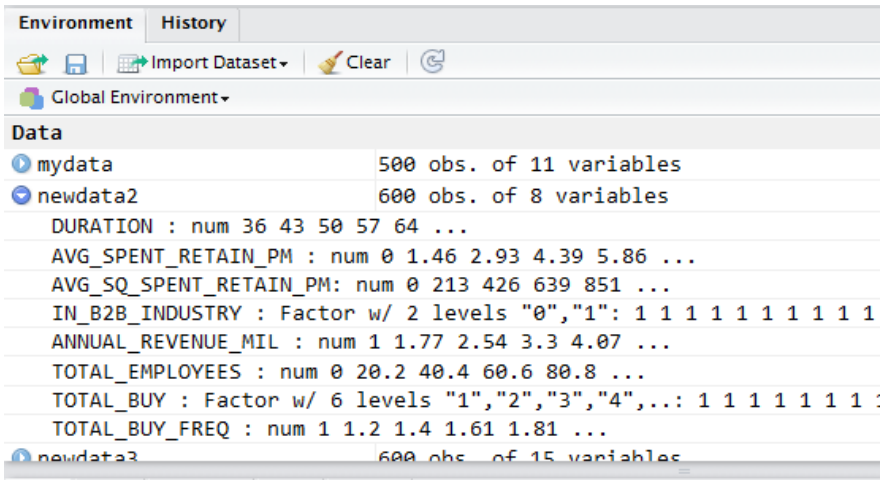
14. Notice that data will be loaded into new structures as the script executes, they will be viewable in the **Environment** tab at the top right are in R Studio:



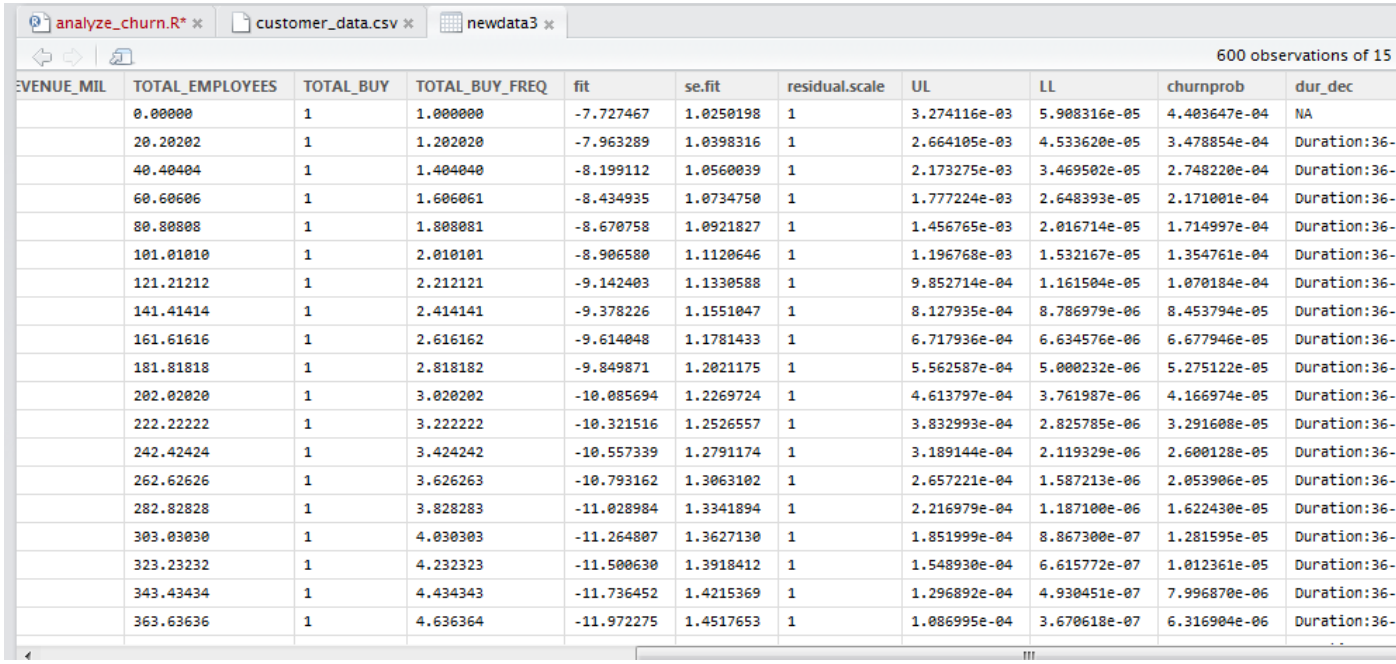
As you step thru the script, notice steps that generate each data item or saved values, for instance:

- mydata <- read.csv("customer_data.csv")
- mylogit <- glm(CENSOR ~ AVG_SPENT_RETAIN_PM …….
- newdata2 <- with(mydata, data.frame(DURATION ……..

15. Any of the data generated can be viewed in summary by clicking on the blue triangle icon next to the item



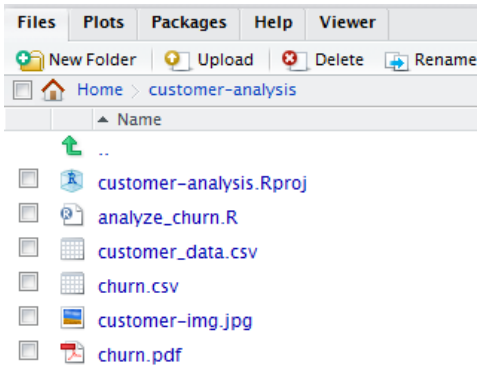
Complete detail can be viewed by clicking on the data object itself, it will open in the editor window, for instance, ***newdata3*** where statistical analysis results are generated:



VENUE_MIL	TOTAL_EMPLOYEES	TOTAL_BUY	TOTAL_BUY_FREQ	fit	se.fit	residual.scale	UL	LL	churnprob	dur_dec
	0.00000	1	1.000000	-7.727467	1.0250198	1	3.274116e-03	5.908316e-05	4.403647e-04	NA
	20.20202	1	1.202020	-7.963289	1.0398316	1	2.664105e-03	4.533620e-05	3.478854e-04	Duration:36-
	40.40404	1	1.404040	-8.199112	1.0560039	1	2.173275e-03	3.469502e-05	2.748220e-04	Duration:36-
	60.60606	1	1.606061	-8.434935	1.0734750	1	1.777224e-03	2.648393e-05	2.171001e-04	Duration:36-
	80.80808	1	1.808081	-8.670758	1.0921827	1	1.456765e-03	2.016714e-05	1.714997e-04	Duration:36-
	101.01010	1	2.010101	-8.906580	1.1120646	1	1.196768e-03	1.532167e-05	1.354761e-04	Duration:36-
	121.21212	1	2.212121	-9.142403	1.1330588	1	9.852714e-04	1.161504e-05	1.070184e-04	Duration:36-
	141.41414	1	2.414141	-9.378226	1.1551047	1	8.127935e-04	8.786979e-06	8.453794e-05	Duration:36-
	161.61616	1	2.616162	-9.614048	1.1781433	1	6.717936e-04	6.634576e-06	6.677946e-05	Duration:36-
	181.81818	1	2.818182	-9.849871	1.2021175	1	5.562587e-04	5.000232e-06	5.275122e-05	Duration:36-
	202.02020	1	3.020202	-10.085694	1.2269724	1	4.613797e-04	3.761987e-06	4.166974e-05	Duration:36-
	222.22222	1	3.222222	-10.321516	1.2526557	1	3.832993e-04	2.825785e-06	3.291608e-05	Duration:36-
	242.42424	1	3.424242	-10.557339	1.2791174	1	3.189144e-04	2.119329e-06	2.600128e-05	Duration:36-
	262.62626	1	3.626263	-10.793162	1.3063102	1	2.657221e-04	1.587213e-06	2.053906e-05	Duration:36-
	282.82828	1	3.828283	-11.028984	1.3341894	1	2.216979e-04	1.187100e-06	1.622430e-05	Duration:36-
	303.03030	1	4.030303	-11.264807	1.3627130	1	1.851999e-04	8.867300e-07	1.281595e-05	Duration:36-
	323.23232	1	4.232323	-11.500630	1.3918412	1	1.548930e-04	6.615772e-07	1.012361e-05	Duration:36-
	343.43434	1	4.434343	-11.736452	1.4215369	1	1.296892e-04	4.930451e-07	7.996870e-06	Duration:36-
	363.63636	1	4.636364	-11.972275	1.4517653	1	1.086995e-04	3.670618e-07	6.316904e-06	Duration:36-

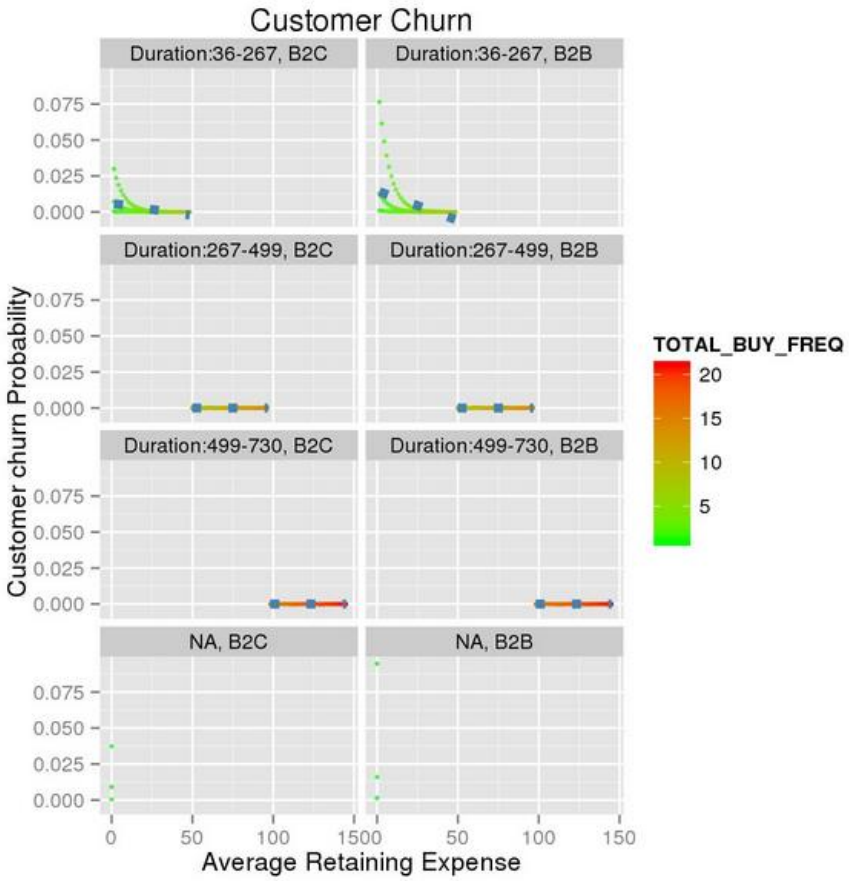
16. Once the script completes (or at each corresponding step for a debug run of the script), output files the script generates from running its analytics functions will appear in your project window in the lower right part of R Studio. Files that this analytics script generates:

- CSV file extract of the customer data and statistical analytics generated
- plotted data in a graphics image file
- corresponding PDF file



17. Click the generated CSV file to view in the editor window - ***churn.csv***

18. Click the jpg plot file – ***customer-img.jpg*** Note that you may need to disable pop-up blocking in your browser

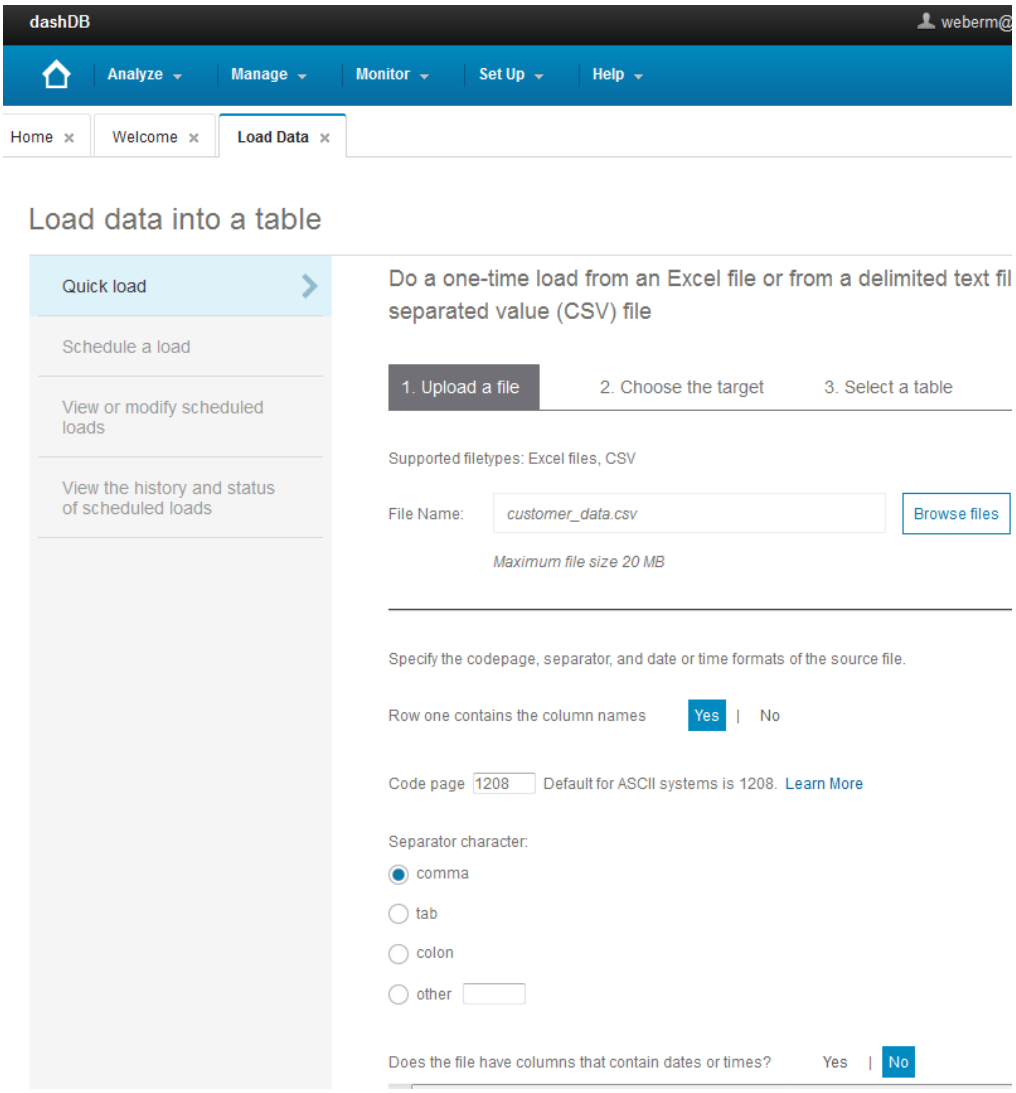


Part 2 – Load and Run the analytics assets in the dashDB analytics warehouse

1. First lets create the data warehouse table from the customer data:

   In the browser window or tab with the dashDB data warehouse console, select **Manage > Load Data**

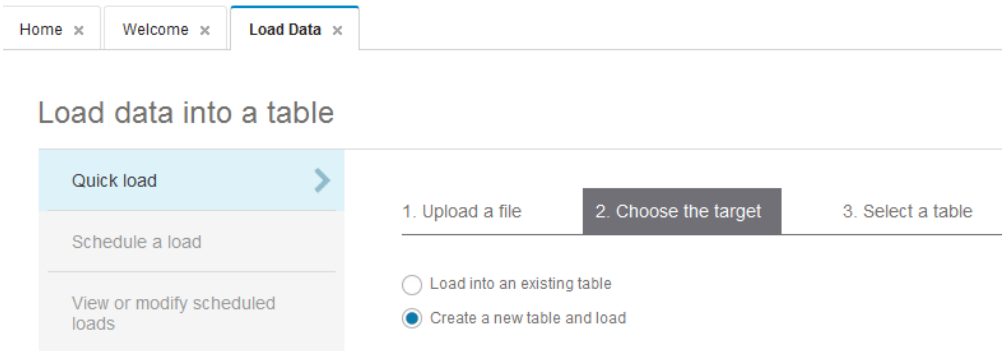
   Use the option for **Quick Load** and browse to select the customer data CSV file as input:



Leave the options for “Row one contains column names”, “Code page”, “Separator character”, and “Does the file have columns that contain dates or times” as default, the file is using a comma delimiter and does not contain dates or times, and the first row does contain column names.

Click **Load**, click **Next**

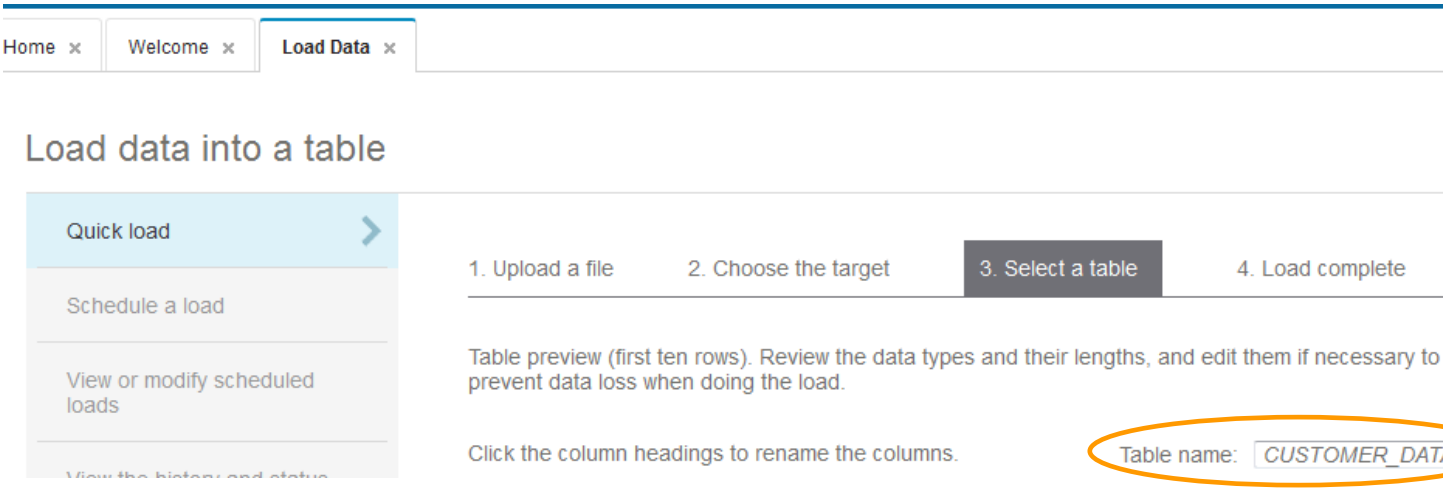
2. On the Quick Load step “2. Choose the target”, select “**Create a new table and load**”. Click **Next**



3. On step “3. Select a table”, In the prompt for **Table name** enter:

   Customer_DATA – make sure the table name is capitalized

   Do not edit or remove any column names/headings, leave as-is, and click **Finish**



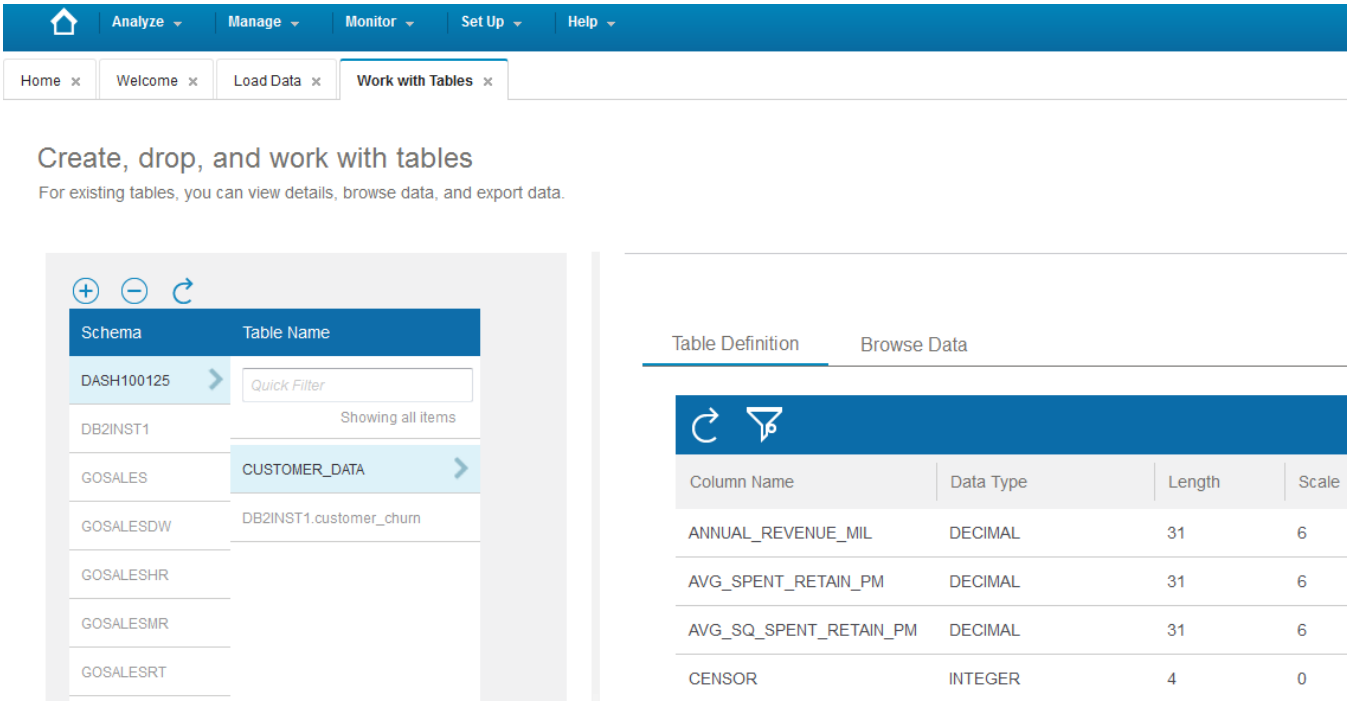
4. There should be a message indicating the load succeeded with the name of the schema the table was created in, ie:

   *Quick load succeeded for table **CUSTOMER_DATA** in schema **DASH100125***

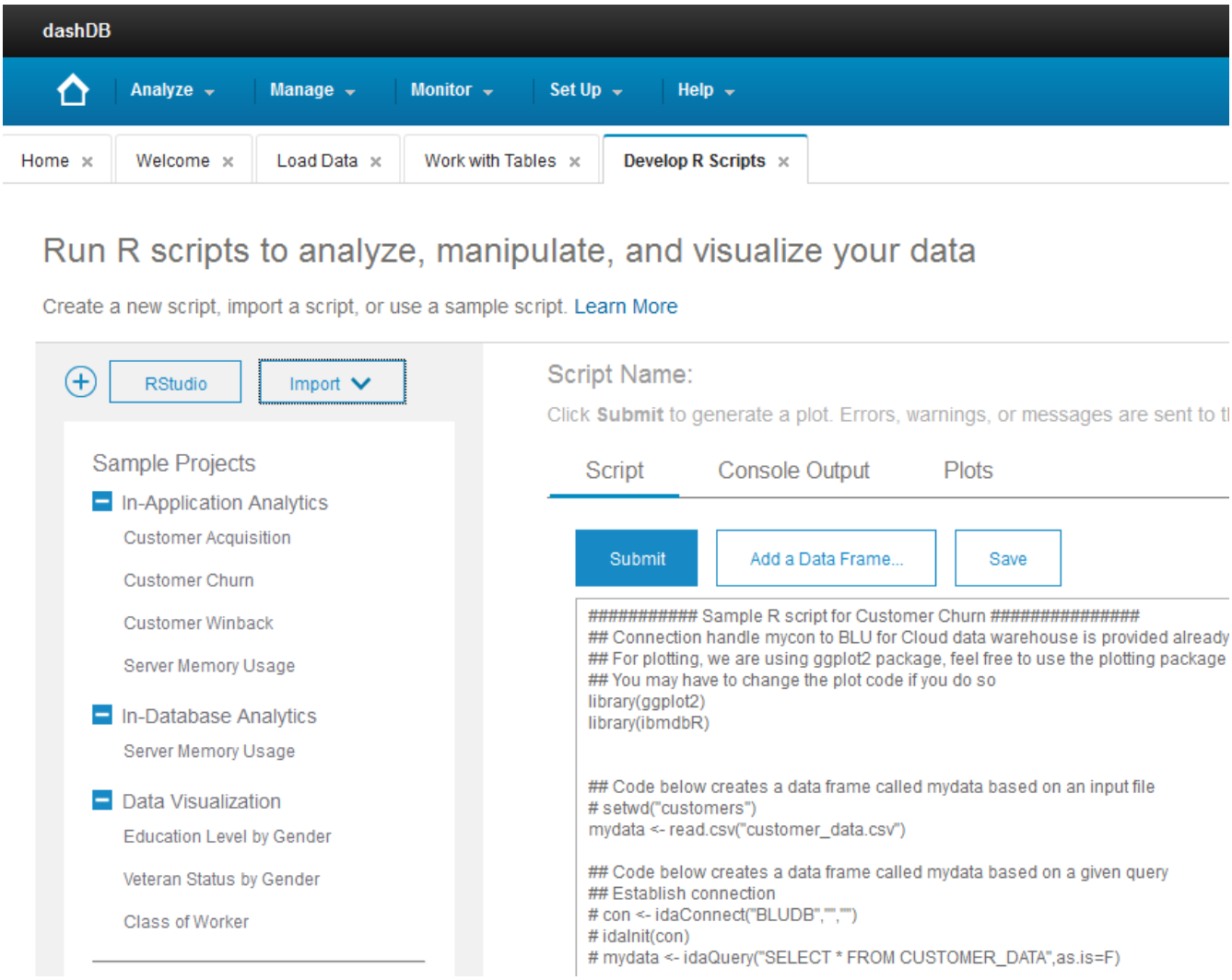
5. In the console, view the database with the table:

   Select **Manage > Work with Tables**

   Click the ***CUSTOMER_DATA*** table you just created, table definition should be shown:



6. In the “**Develop R Scripts**” tab of the dashDB console, click **IMPORT > Import from local filesystem > Browse** and select the R Script you downloaded, it will be loaded into the script editor:



7. The script is currently configured to use input from a local CSV file for the customer data, it must be changed to connect to the dashDB data warehouse and use the table you created:

   Comment (# character) the line that reads the data from the CSV file:

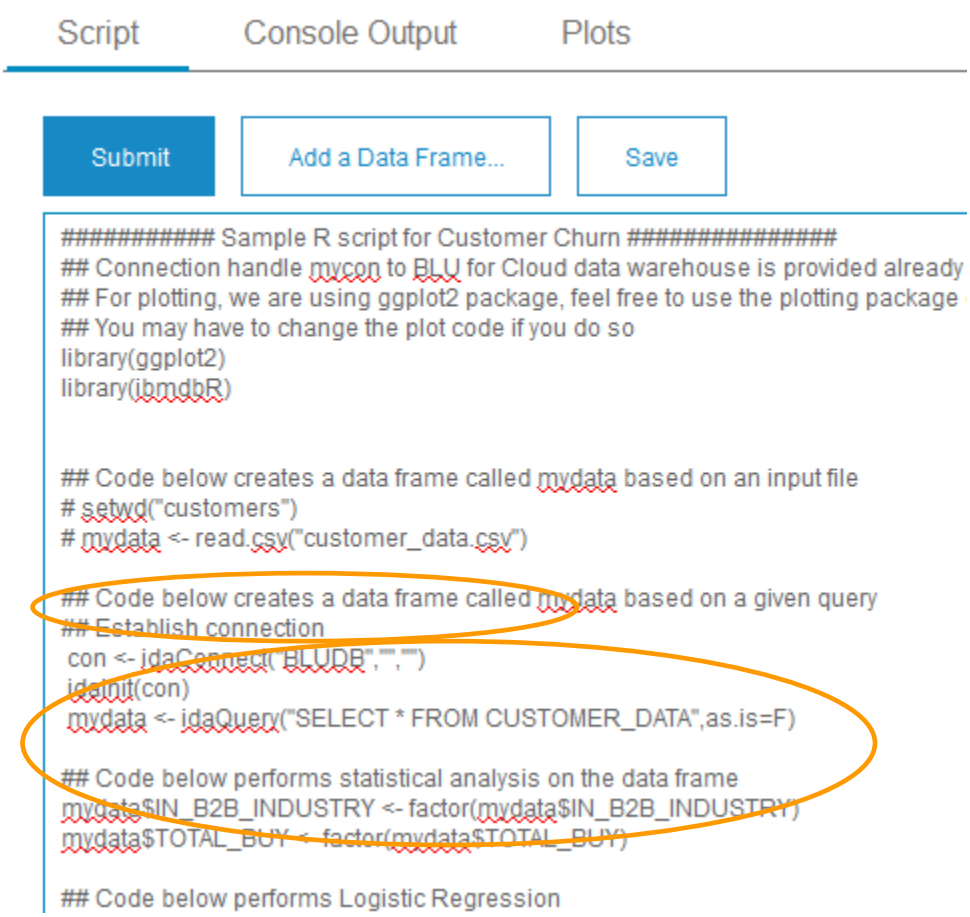
   ```
   # mydata <- read.csv("customer_data.csv")
   ```

   Uncomment the lines that establish the database connection and reads the data from a query:

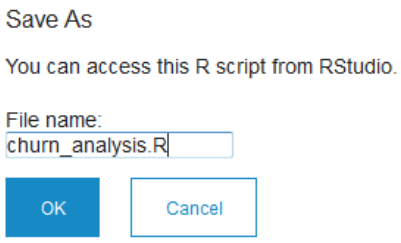
   ```
   con <- idaConnect("BLUDB","","")
   idaInit(con)
   mydata <- idaQuery("SELECT * FROM
   CUSTOMER_DATA",as.is=F)
   ```

   Scroll down to the end of the script and uncomment the line that closes the database connection:

   ```
   idaClose(con)
   ```

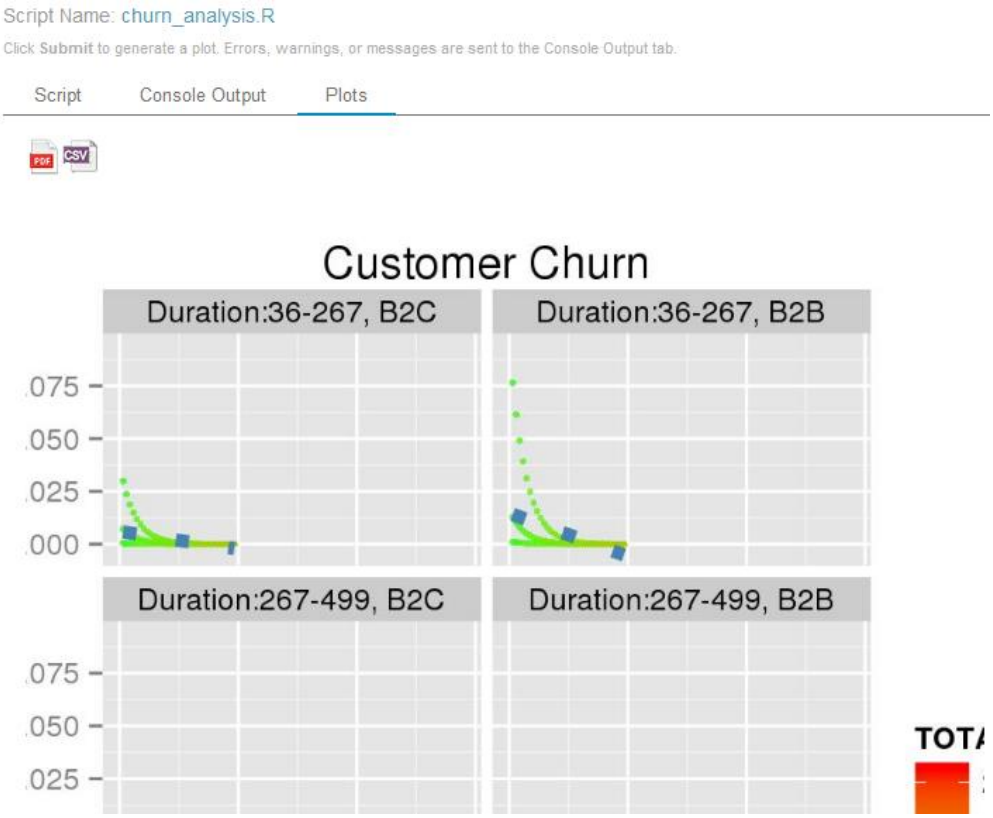


Click **Save** and provide a name for the R script, making sure to name the script with a ".R" extension



Click **OK**

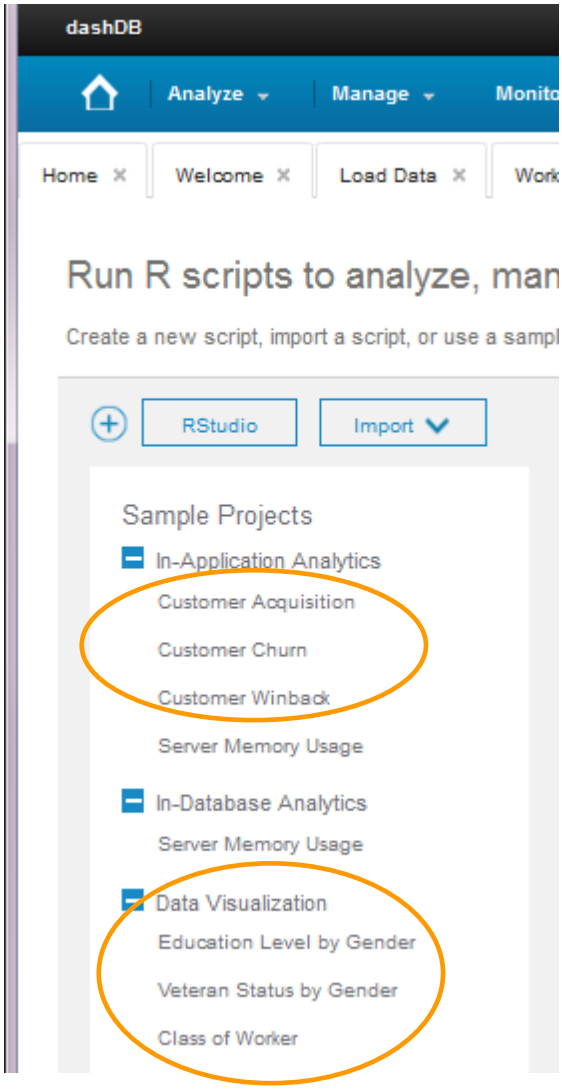
8. Select **Submit** to run the script and examine the analytics results in plot format – the x axis plots the expense of customer retention and the y axis plots the probability of customer churn. Subsets of customers are groups for duration and B2C/B2B customers:



You may have to scroll the browser window view to see the full plot or the plot color coding key for Total Buying Frequency

Notice also that the script once again generated the CSV file with the customer data plus the statistical analysis results, and it also generated the PDF file.

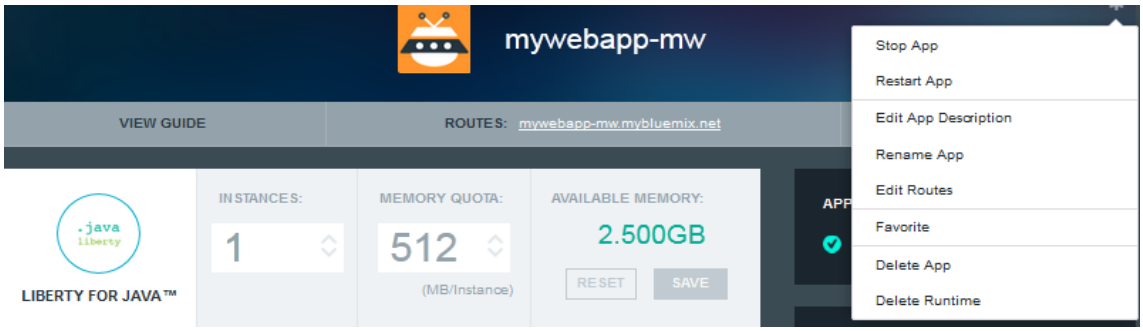
9. Sample scripts and data are included for several business analytics scenarios, including customer insight and demographics – you may want to get started working with these, or for working with analytics R language scripts, resources are provided in the section below, after step 10.



Clean-up your applications and services in Bluemix (optional):

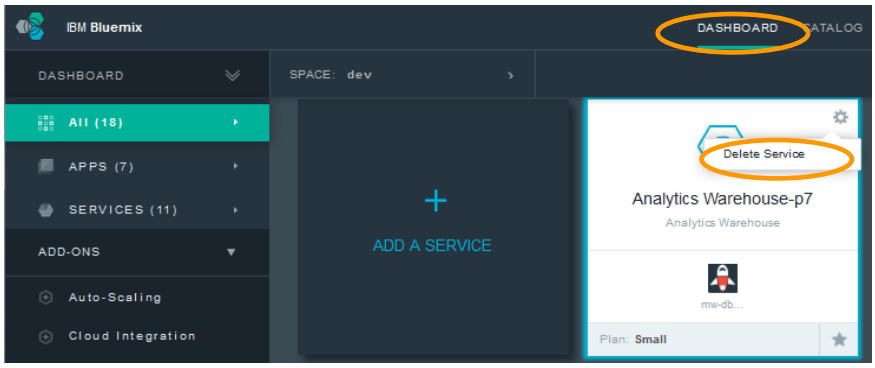
10. You may want to make sure to delete the application(s) you have been working with in this exercise, as well as the services they are bound to, in order to free memory and resources for use with other applications and learning exercises you may be working with later:

    In the Bluemix **Dashboard,** select your application, and in the Application Overview, you can use the application control widget in the top right corner of the overview to delete the application- Click on **Delete App**



    When prompted if you also want to delete the included services bound to the application select **OK.** If you are prompted to delete any routes for your application, also press **OK.**

If you are not sure whether service instances were deleted, you can view all applications and services in your Bluemix space in the Bluemix **Dashboard**, scroll to the **Application Services** section and use the control widget on any individual service instance to delete – click on **Delete Service**



Guides to additional resources to get started with R Analytics and the dashDB data warehouse service:

- Beginner's guide to R
- R Analytics homepage – includes links to download R Studio, documentation, R language conference submissions, tutorials, replays, newsletters and user groups
- DataCamp – R tutorials and data science courses
- Big Data supercharges enterprise analytics with R
- Free online course Introduction to Data Analysis Using R at Big Data University !
- dashDB for Bluemix powered by BLUAcceleration and Netezza data warehouse appliance – documentation
- dashDB analytics data warehouse – resources and getting started

Congratulations *You have now completed the "Big Data & Analytics Exercises – Analytics for Hadoop (BigInsights) and Analytics with Data Warehouse" in Bluemix including building and running a Java BigData application, working with the BigInsights Hadoop service management console, and working with the R analytics and the dashDB data warehouse. You can take advantage of many resources available to maximize your exposure to technical education and skills building, explore the many Bluemix applications and services available, and participate in opportunities to collaborate with other Bluemix developers in the Bluemix ecosystem:*

Next Steps – Explore Bluemix further and participate in the Developer Ecosystem

Visit the Bluemix Developer Community at IBM developerWorks to get started using Bluemix with self-service documentation, tutorials, sample projects, articles, and workshops. Use the Bluemix developer forum and blog to get answers and follow our blog. Participate in Events listed in the calendar including Bluemix and Cloud related workshops, conferences, meetups and technicall briefings
https://developer.ibm.com/bluemix/

Dev2Dev Support: Community-based direct support network by linking developers to other developers:
https://www.ibmdw.net/bluemix/

Take advantage of technical guidance and resources highlighted in Our Bluemix Days Technical Enablement team - Blog

IBM Open Source on GitHub – the new IBM@GitHub OSS portal, aggregating all IBM OSS projects on github
http://ibm.github.io/

IBM DevOps Services, powered by JazzHub, where you can collaborate with others to develop, track, plan and deploy software. Share your public projects, or manage your work in private projects. Registration is quick and free. DevOps services on JazzHub provides Git hosting, built-in Web IDE, integration with Eclipse, Visual Studio, or your tool of choice. Automatically build and deploy your application to IBM's cloud platform, Bluemix. Use Team Collaboration and share your work and collaborate through expert tools for Agile Development.

http://hub.jazz.net

Recommended articles and resources for further exploration with Bluemix:

- Getting started with BlueMix DevOps: https://hub.jazz.net/tutorials/jazzeditor

- Bluemix projects repositories using a variety of runtimes and application services in Bluemix to quickly get started with example applications: https://developer.ibm.com/bluemix/docs/sample-code/

- More advanced Java EE application based on the "Trade" enterprise application benchmark:
https://www.ibmdw.net/bluemix/2014/04/18/complex-java-ee-application-bluemix-cloudtrader/

- Mobile list management in the cloud using Mobile Web Starter boilerplate:
http://www.ibm.com/developerworks/library/mo-android-mobiledata-app/index.html

- Mobile marketing application with the Mobile Web Push service:
http://www.ibm.com/developerworks/library/mo-push-engage-app/index.html

- Hackathon Starter Project and DevOps: http://www.ibm.com/developerworks/cloud/library/cl-hackathon-app/index.html

- Build a data mining app in Bluemix using Java, WEKA and the Analytics Warehouse service:
https://www.ng.bluemix.net/docs/#services/AnalyticsforHadoop/index.html#analyticsforhadoop

- Getting started with Big Data Analytics with InfoSphere BigInsights in Bluemix powered by Hadoop:
https://www.ng.bluemix.net/docs/#services/AnalyticsforHadoop/index.html#analyticsforhadoop