Build a Shiny application to analyze #Bluemix sentiment using the Bluemix R custom buildpack

Associate Professor of Marketing at Gonzaga University
Gonzaga University

IT Specialist Big Data and Analytics

Create a Shiny application that analyzes the popularity of entities on Twitter and performs a sentiment analysis of the tweets using R. Learn how to run R on IBM® Bluemix® using a custom buildpack. Customize your application to perform any kind of complex analysis on your data stored in the cloud.

Create an R-based application to measure the popularity of two different entities on Twitter and perform a sentiment analysis of their tweets. Get a glimpse of the techniques to determine popularity, and find out how to create new applications on IBM Bluemix that use R.

Sign up for IBM Bluemix™
This cloud platform is stocked with free services, runtimes, and infrastructure to help you quickly build and deploy your next mobile or web application.
"Use all the power of R analytics on Bluemix to compare the popularity of Twitter hashtags. It's easy with the R custom buildpack on Bluemix!"

The application uses the Shiny server framework to create an interactive, web-based method to visualize the results. It uses a custom buildpack to run R on Bluemix. The popularity of R (ranked by IEEE Spectrum as the #9 programming language) has recently increased significantly. Learn how to use its power on Bluemix.


What you'll need to build your application

- A Bluemix account
- A DevOps Services account linked to your IBM ID
- A GitHub account
- Access to the Twitter API
- Familiarity with R
- RStudio with version control with Git and SVN

Run the app
Get the code

Step 1. Fork the application from IBM DevOps Services

Click Get the code to export the source from IBM DevOps Services:

1. Log in to DevOps Services.
2. Click **Edit Code**.
3. Click **Fork > Fork with a new JazzHub project > Save**.

### Step 2. Connect RStudio with IBM DevOps Services

Open RStudio with Version Control with Git installed by using the following steps.

1. **Click Project > Create Project...**
2. **Click Version Control > Git.**

3. Enter the URL of the repository on DevOps Services [https://hub.jazz.net/git/crackmanworld/TwitterBluemix](https://hub.jazz.net/git/crackmanworld/TwitterBluemix). The project name is automatically selected.
Enter the directory where you want to save the project on your computer.

Now your project is created in RStudio. You can use your R skills to modify the application or create a new similar Shiny application.

Step 3. Push the application to Bluemix using a custom buildpack

The Bluemix catalog includes more than 15 open-source, Cloud Foundry-compatible buildpacks.

For this application, use a buildpack that is developed for Heroku but works well for Bluemix. Use the Bring Your Buildpack option. The code has been modified to use a recent version of R, R 3.1.

1. From the command-line interface, log in to Bluemix using the following code as an example.

```
C:\>cf login
API endpoint: https://api.ng.bluemix.net
Email> armand.ruiz@fr.ibm.com
Password>
Authenticating...
OK
Targeted org armand.ruiz@fr.ibm.com
Targeted space dev
API endpoint: https://api.ng.bluemix.net (API version: 2.4.0)
User: armand.ruiz@fr.ibm.com
Org: armand.ruiz@fr.ibm.com
Space: dev
C:\>
```
2. Specify which custom buildpack is going to be used:

   > cf push [NEW_APP_NAME] -b [GIT_REPO_URL]

3. For this application the [GIT_REPO_URL] is:

   https://github.com/aruizga7/cf-buildpack-r

4. Navigate to the directory where your R Shiny application is stored and push the application to Bluemix with the R custom buildpack.

   C:\TwitterBluemix>cf push MyNewShinyApp -b https://github.com/aruizga7/cf-build
   Creating app MyNewShinyApp in org armand.ruiz@fr.ibm.com / space dev as armand.ruiz@fr.ibm.com...
   OK
   Creating route mynewshinyapp.mybluemix.net...
   OK
   Binding mynewshinyapp.mybluemix.net to MyNewShinyApp...
   OK
   Uploading MyNewShinyApp...

Expect to see many log messages during the installation of R and all the R packages that this specific application is using.

**Step 4. Twitter analysis with R**

The application compares the popularity of two entities on Twitter and performs a sentiment analysis of the tweets of the two entities. The estimate of popularity uses two methods based on algorithms borrowed from Jeffery Stanton, Jeffery Breen, and Gaston Sanchez.

- The application determines the probability of a new tweet regarding an entity occurring within a certain time period. It is assumed that the higher this probability, the higher the frequency of discussion of this entity on Twitter. Higher frequency suggests relatively higher popularity.

  READ: Jeffrey Stanton’s book on Data Science

- The application compares the proportion of all retrieved tweets for each entity that occurred within a certain time frame. Next, it determines the extent to which the discussion on Twitter is positive or negative. Words used in tweets are matched with a list of terms deemed positive or negative, based on previous research. A score is then generated for each tweet, based on the number of positive and negative terms used.

  READ: Jeffrey Breen’s work on text mining

  READ: Gaston Sanchez’s work on sentiment analysis

**User-specified input to the application**

- **Entity 1** and **Entity 2**: The default entities are Bluemix and Heroku.
- **Number of tweets to retrieve**: This application uses the R package twitteR, which uses a public API. To keep Twitter from blocking the application, the maximum number of tweets is restricted to 50. To modify the value, simply change the code.
- **Time (in seconds)**: Sets the interval used to calculate the popularity. The calculation is based on the proportion of tweets for both entities that arrived within a particular time period.
Results output from the application

The results are presented in seven tabs.

- Tab 1: Shows how many tweets were retrieved and plots the probability of a new tweet arriving within a particular time for both entities.

![Number of tweets](image)

![Probability of a new tweet arriving within a particular time, t](image)
• Tab 2: Includes three graphs that visualize the distribution of delay times between tweets for both entities. The first graph gives box plots and the mean of the distribution of delay times for both entities. The second and third graphs are histogram and kernel density function.

• Tab 3: Shows a bar graph of the estimated proportion of tweets retrieved for both entities for delay times greater than the user-specified time. It also plots the 95 percent confidence interval in the estimation of these proportions.

• Tab 4: Shows the sentiment analysis of the tweets. The distribution of sentiment scores (a higher score shows greater popularity) for the retrieved tweets for both entities is presented using box plots. The mean is also computed.

• Tab 5: Presents word clouds of terms used in tweets for both entities.
• Tab 6 and 7: Presents the raw tweets retrieved for the two entities and used in the outputs generated in the other five tabs.

**Step 5. Customize your application**

*R* is a powerful and flexible data analysis language and platform. A variety of **packages** enable you to perform any kind of analysis you can imagine. Install new packages by adding them in the *R* file `init.R`, which is run when you push the application. Add code to install any package you need. Use the following code as an example.

```
# Example `init.r` file
install.packages("twitteR", clean=T)
install.packages("stringr", clean=T)
install.packages("ROAuth", clean=T)
install.packages("RCurl", clean=T)
install.packages("ggplot2", clean=T)
install.packages("reshape", clean=T)
install.packages("tm", clean=T)
install.packages("RJSONIO", clean=T)
install.packages("wordcloud", clean=T)
install.packages("gridExtra", clean=T)
```

If you know how to program in *R*, it's easy to get started with the Shiny framework; check out the many tutorials available.

**READ:** [Getting started with R](#)

**READ:** [Getting started with Shiny](#)

**READ:** [Getting inspired](#)

**Conclusion**

With *R*, perform any kind of analysis of data from a variety of sources, such as live data like tweets or data stored in a database or in a flat file. Use the Shiny framework to make your analysis available from interactive web applications. Thanks to Bluemix, you can deploy and distribute these applications easily, make use of all the services available, and provide worldwide access in seconds.

Now that you know how to use the *R* custom buildpack to do a simple analysis of tweets, you can apply the power of Bluemix analytics to other applications. Leave us a comment to let us know how it goes!

**READ:** [Read Armand's blog](#)

**RELATED TOPIC:** [R programming language](#)

**RELATED TOPIC:** [Sentiment analysis app with Node.js, Express, sentiment, and nTwitter](#)
About the authors

Vivek Patil
@vivekhpail on Twitter
Follow me on Google+

Armand Ruiz Gabernet
@armand_ruiz on Twitter

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