Choice of Visualization Product

Rani Fields

For our project, all data is static and incremental uploads will take place every year to 2 years. We do not need to take into consideration streaming data, high velocity or data approaching ‘big data’ status at the moment. We as a team decided that the best way to output a response to the RQ would be to create a dashboard and story documenting our analysis. The audience we foresee reviewing this analysis will be the general public and our stakeholders.

Tableau Server

We initially explored Tableau Server as an easy to access solution which facilitates collaboration. While this solution is costlier than other solutions we covered due to both hosting and licensing costs, the familiar UI would allow for rapid dashboard development and the unique workbook management system would allow us to work on our visualizations in tandem with various team members. Tableau Server has a stringent list of hosting requirements. Notably, Tableau Server required Windows 2012/2016 Server, two physical cores, and 15 GB of RAM[1].

While Tableau Server would have been the best solution despite the cost, issues associated with the installation process preventing us from pursuing this route.

R Shiny

R Shiny requires a small amount of time to learn how to code if you have experience with R. Due to ample tutorial resources, it should take the average R user no more than 2 days to learn the system. if you lack R knowledge then it could take weeks. For this project, however, all team members should be versed in R which is why this solution was considered.

If working on a team that does not code in R, Plotly, Bokeh, Seaborn are other considerations for those who code in Python, while D3 visualizations take the same amount of time if able to create JavaScript, HTML, CSS. There is also a myriad of other strictly JavaScript solutions such as HighCharts and Chart.js. What these solutions boils down to is building up a visualization or dashboard from code and packages. It requires one to know enough in one of the above languages as well as the statistical knowledge to create a dashboard or visualization. Additional time and effort comes from coding each visualization and having to establish the composition of the dashboard with care, however, once a baseline is established components can be copy and pasted. Despite the issues with programmatic databases, we found it useful to attempt to use Shiny due to its collaboration potential.

---

[1] “Shiny Server Professional v1.5.3 Administrator's Guide”

Our team spent 48 hours working with both Tableau Public and R Shiny to create a base dashboard with visualizations to see what would be the best option for this current project. In R Shiny we were able to create two base functional dashboards with limited visualizations each, while we were able to create a multitude of dashboards in Tableau Public.

Tableau Public

Tableau Public for our team provided an easier solution for the creation of visualizations and dashboards. We were able to create three dashboards and over ten visualizations. Tableau won in speed of creation over R Shiny. Additionally, in a real world setting we would be able to create new visuals quickly and easily to present to stakeholders. R Shiny would be able to create more detailed visuals with more options for statistical analysis. If this was not a project we would take into consideration in more detail what our stakeholders were requesting and if the analysis required more advanced statistics we would most likely move to R Shiny or create an additional layer (possibly presentation layer) in R Studio before outputting the data to Tableau.

Tableau Server vs. R Shiny: a Detailed Look

As Tableau Server and R Shiny were our primary solutions, we've compiled a detailed look into the differences between the two technologies. The primary areas we will be covering are set-up and maintenance issues, the costs associated with each product, the flexibility of each product, and each product's conduciveness to advanced analytics and automation.

On the next page you will find a table documenting each one of these key areas in detail. Whichever product appears to offer the stronger solution for that area will be color-coded green. Please note that we are judging each area strictly in the context of our project. That is, we weigh time-related risks with disproportionate importance.

A point-by-point look at R Shiny vs Tableau Server

Area	R Shiny	Tableau Server

Set-up / Maintenance	• Likely will need to deal with R dependencies • R updates can break Shiny • Low set-up time unless issues with R arise	• Difficult to set up • Tableau Server stability and maintenance risks are currently unknown however individual reviews[2] imply Tableau products to be low risk
Costs	• Open source • An accessible free version is available • Commercial solutions are available if certain features are required	• $~~300~~293/mo for around-the-clock hosting, does not include EBS hosting. Includes licensing (5 users) and VM costs. • A more detailed cost analysis can be found below.
Flexibility	• More options for bringing in multiple sources • Creating visualizations is not as simple as drag-and-drop • Can copy template widgets with ease: just copy and paste!	• Cumbersome UI makes simple tasks difficult • UI-based dashboard design allows for rapid development
Advanced Analytics & Automation	• Dashboards are engineered in R, a premier analytical language • Programmed dashboard: allows for extensive automation, especially with regards to complex analysis	• Large in-house suite of analytics features such as k-means clustering and zip code handling • Recently added Python support[3] • Due to the recently added Python API, more analysis needs to be performed to gauge advanced analytics and automation ability.

At this time, we cannot make any claims about load and performance. We do know that both Tableau Server and Shiny (open source) do not have a cap on concurrent dashboard viewers.

Tableau Server: Financial Analysis

In this section we will briefly analyze Tableau Server's potential costs. As other solutions are low-cost options and the Tableau Server option has the potential to become unreasonable costly, extreme care was taken in ensuring we chose only the instance type required. The primary source of cost is from the hardware and OS requirements for Tableau. As of the start of our project, Tableau Server mandated a Windows OS. Tableau also required two physical cores, but it might be possible to work with virtual cores in lieu of physical cores. Experimentation is required to see how Tableau behaves in this situation. Tableau also required a number of other specifications, but the remaining specifications (e.g. RAM) were found to not hold a large financial footprint.

---

[2] "Review of Tableau by a Real User" by Bhosale, Dev

[3] "Leverage the power of Python in Tableau with TabPy" by Beran, Bora

In the table below, we provide three columns of interest: Cost (24-hr), Cost (12-hr), and Cost (8-hr). In order to limit costs, we were looking at the option of only running the host VM for a limited number of hours a day. Enabling and disabling the VM would be handled by a lightweight script. Please note that one team member has extensive experience in creating such scripts so knowledge and skill were not an issue.

Windows VM costs, 8/20/2017, all costs are 30-day approximates assuming the VM runs for x hours a day. All data was pulled from the official AWS site.

Instance	Risks Associated	Cost (24-hr)	Cost (12-hr)	Cost (8-hr)
t2.xlarge	• Requires EBS for storage	$164.88	$82.44	$54.96
m4.xlarge	• Requires EBS for storage • Insufficient physical cores, unknown issues likely	$276.48	$138.24	$92.16
m4.2xlarge	• Requires EBS for storage • Requires strict active hours management to limit costs	$552.96	$276.48	$184.32

Summary

We chose Tableau Public over R Shiny due to time constraints and to build off of existing, functional, and refined work done in Tableau. This became a boon for us: due to this change, we were able to find a number of DQ issues through our EDA with Tableau Public.

Works Cited

1. Gupta, Anand. “R Shiny v Tableau: Dawn of Graphics.” LinkedIn, 25 Apr. 2016, https://www.linkedin.com/pulse/r-shiny-v-tableau-dawn-graphics-anand-gupta Accessed 21 Aug. 2017.
2. Beran, Bora. “Leverage the power of Python in Tableau with TabPy.” *Tableau Software*, Tableau Software, 4 Nov. 2016, https://www.tableau.com/about/blog/2016/11/leverage-power-python-tableau-tabpy-62077. Accessed 21 Aug. 2017.
3. Bhosale, Dev. “Review of Tableau by a Real User.” Unbiased reviews from the tech community, 22 Nov. 2016, https://www.itcentralstation.com/product_reviews/tableau-review-37875-by-dev-bhosale Accessed 22 Aug. 2017.
4. Jain, Rashmi. “A beginner's tutorial on the apriori algorithm in data mining with R implementation.” *HackerEarth Blog*, HackerEarth, 3 Mar. 2017,

https://www.itcentralstation.com/product_reviews/tableau-review-37875-by-dev-bhosale. Accessed 21 Aug. 2017.
5. “Shiny Server Professional v1.5.3 Administrator's Guide.” Shiny Server v1.5.3 Configuration Reference, http://docs.rstudio.com/shiny-server/#system-requirements. Accessed 21 Aug. 2017.
6. “Amazon EC2 Instance Types – Amazon Web Services (AWS).” Amazon Web Services, Inc., https://aws.amazon.com/ec2/instance-types/. Accessed 20 Aug. 2017.
7. “EC2 Instance Pricing – Amazon Web Services (AWS).” Amazon Web Services, Inc., https://aws.amazon.com/ec2/pricing/on-demand/. Accessed 20 Aug. 2017.
8. “Is there a limit on the number of concurrent connections per server for Shiny Server?” RStudio Support, https://support.rstudio.com/hc/en-us/articles/218294957-Is-there-a-limit-on-the-number-of-concurrent-connections-per-server-for-Shiny-Server-. Accessed 22 Aug. 2017.