I am very excited to have this chance to introduce R Shiny programming and its application to our biostatistical team. It is very difficult even impossible to talk about R shiny programming technical details in 20 minutes. So today I will just introduce the basic ingredients of R shiny programming and focus on introducing its application in pharmaceutical / device industry and illustrating how to develop my first R shiny application for sample size calculation.

Research managers, CAPM and other function colleagues always come to our biostatisticians consulting some questions, for example, sample size calculation. In this slide, I will give a hypothetical example that our biostatisticians are very familiar with.

Research manager asked biostatistician:” how many subjects are at least required for a MR trial based on China NMPA guidance?”. Biostatistician answered: “At least 54 subjects”. RM continued:” But our budget is limited, is there any possibility to reduce the sample size?”. Bios answered: ”If the expected proportion of acceptable image sets is above 90%, sample size is smaller than 54. Can you provide me the expected proportion?” RM continued: ”I only know our expected proportion is ranging from 91% to 96%. Can you help me calculate the corresponding sample sizes?”. Bios answered: “Ok. The bigger expected proportion is, the smaller the sample size is. I will send you the sample sizes by email.”

After the conversation, biostatistician wrote a SAS macro to calculate the sample sizes based on the different expected proportions. One hour later, biostatistician sent the sample size table to research manager and the RM was very happy to see this table. For this simple calculation, Biostatistician can calculate the sample sizes quickly. For the complicated cases, such as, the expected proportion ranging from 0.01 to 0.99, Biostatistician need to produce a very long table to demonstrate the sample sizes. Is there any more efficient method?

Sample size is like a fish. Give research manager a sample size table and you meet his need for one time. Teach RM how to calculate the sample size and you meet his need on sample size calculation for longtime. What is fishing tool to fish sample size? SAS macro or R function for sample size calculation, is one fishing tool to fish sample size, but it is difficult to teach RM to learn SAS or R? It may be very cool if we can develop an interactive webpage and say, “here you go it’s an interactive webpage for sample size, you can alter the parameter based on your need and sample size will appears right away in screen.” R shiny is an interactive webpage fishing tool for RM to calculate the sample size without requiring any technical and software background knowledges.

Now let us look at the fishing tool for RM, the rshiny web application for sample size calculation. In the left of webpage, there are two options to choose, MR and Ultrasound. Choose MR and then four parameters are listed in below. The first parameter is expected proportion, the second parameter is target proportion, the third is type I error and the fourth is power. We set the four parameters by default based on the suggestions in NMPA, and the result is 53.89 as shown in right of webpage. Once one of the numbers for the four parameters changed, the sample size result will be altered right away accordingly. This interactive and dynamic application is easy to use for RM. RM will be much happier to see the Rshiny web app for sample size calculation.

Through this hypothetical example, we have a general idea about what R shiny can do. What is R shiny? Shiny is an open source R package that provides an elegant and powerful web framework for building web applications using R. Shiny helps you turn your analyses into interactive web applications without requiring HTML, CSS, or JavaScript knowledge. Shiny can help statistician and programmers automate their route tasks with great efficiency, such as, sample size calculation, datasets visualization. R Shiny is similar to SAS macro but it is more dynamic, interactive and powerful. It is easy for statistician and programmers to develop interactive data visualization in web and make it accessible to other team members. R shiny is very easy to use even for non-programmer. It is a great interactive and powerful tool for the collaboration between statistician and other function team members.

As a dynamic, interactive and powerful tool, Rshiny is widely used in pharmaceutical industry. Recently, R shiny is also used by FDA reviewers to streamline their statistical review work.

Let us look at some live shiny examples in pharmaceutical industry, Marc Lavielle built a PK model using R shiny. As shown in the right screenshot, the user can produce the PK plot easily by inputting choosing the parameters. Andrey developed a shiny app for bioequivalence. FDA developed an AE dashboard using R shiny.

R shiny is also used by statisticians in medical device industry. For example, Roche diagnostics team develop a rshiny app to create validated reports for regulatory submissions. bioWARP enables people using advanced statistical methods, who cannot program R. It builds a connection to the validated R-packages developed at Roche with an easy to use and elegant user interface. Its most important feature is the ability to move all statistical evaluations right into PDF reports. These are validated and can directly be used for submission to regulatory authorities. bioWARP is called the “largest shiny application in the world” by us as it already consists of 16 tools, has over 100.000 lines of code, >500 buttons and interaction items and is growing and growing and growing. But this shiny app is not accessible to the users outside of Roche diagnostics.

Almost all the statistical work our biostatistician can imagine, from trial design to statistical analysis, can be done by developing shiny app. In the following, I will demonstrate how to develop a shiny app using an example on the sample size calculator I developed.

This is my first R shiny app – medical device sample size calculator, which can be used to calculate sample size for MR and Ultrasound. We can extend this app to include other medical device products, such as, CT and other devices.

There are two steps to develop a R shiny web app. First step is to write R shiny programming code and the second step is to publish shiny apps on the web with shinyapps.io. How to write R shiny programming code? Open R studio IDE firstly, and then open the new file and choose shiny web app to start writing shiny programming code. Once code is ready, publish shiny app with shinyapp.io account on the web.

There are three core ingredients for the Shiny app programming. Firstly, define user interface and this part determines the layout in the final shiny app web. Secondly, define server function and this part is based on the calculation algorithm, for example, the sample size calculation formula. Thirdly, run the application code.

Now let us look at the user interface. This is the whole user interface programming code, the first part is for the general information. The second part is for calculator. Now let us take a look at the server function. The first part is based on the below sample size formula. The second part is for the layout of sample size in shiny web app.

After R shiny programming code is ready and the results are validated by another programmer or statistician, it is time to share the shiny app online.

The easiest way to turn Shiny app into a web page is to use [shinyapps](http://www.shinyapps.io/) account, RStudio’s hosting service for Shiny apps. Shinyapps.io lets user upload the app straight from R session to a server hosted by RStudio. Firstly, user needs to create a shinyapps.io account.

Once you have an shinyapps.io account, shinyapps.io automatically generates a token and secret for you, which the rsconnect package in R Studio can use to access your account. Retrieve your token from the shinyapps.io dashboard by selecting the Tokens option in the menu at the top right of the shinyapps dashboard (under your [avatar](https://en.gravatar.com/)). Click the Show button on the Token page. A window will pop up that shows the full command to configure your account using the appropriate parameters for the rsconnect::setAccountInfo function.

Copy this command to your clipboard, and then paste it into the R console in the RStudio IDE and hit run.

After running the rsconnect code successfully, it is time to publish R shiny app to webpage. Hit the “publish application” at the top right of R Studio IDE and then the below screenshot will appear. Now you need to connect publishing account, press the next and choose the shinyapps.io and then copy the Token information to paste here. After connecting account successfully, you can publish the app successfully to the server. However, it fails to connect to shinyapps.io account using GE laptop. So I use my own laptop to connect my shinyapps.io account and publish sample size shiny app to server. This screenshot shows that Rshiny account named kangrinboqe is destination account and I can publish R shiny app into this account. The final step is to hit publish.

After publishing rshiny app successfully, now in my account we can see the status is shown as running, the url and active hours are shown here. We can use shiny freely, but the free shiny has some limitations: at most 5 applications can be developed and 25 active hours used for free users. More pricing information on Rshiny, please look at the below screenshot. Users can go for professional server that offers very well secured and scalable server environment. In this slide, a screen recording will show us how to develop a shiny web app vividly.

Open R studio IDE, create a shiny web app. This is the shiny example that R studio provides. This is the rshiny code for sample size calculation. Once code ready, hit Run App. Now we enter into the app locally. Now hit publish and share it into web. Firstly we need to log into the rshinyapps.io account. Copy the token information into R studio IDE and run the code using package rssconnect. Now the R IDE is connecting account successfully. And then hit publish, wait for a while. Now the R shiny app is successfully published into rshiny account. Anyone can enter into the webpage and calculate sample size freely now. In the account, is shows a shiny app is at running stage.

Thank you for listening to my introduction of R shiny, I hope in the future our biostatistical team can develop some R shiny applications based on the needs in our daily work. I think R shiny will help us communicate with other function colleagues more efficiently and make some boring daily work interactive and cool.