

Linux服务器部署Shiny Server

Linux Shiny Server

- 此文档要求系统Ubuntu 12.04及以上。

1. 下载R

```
$sudo apt-get install r-base
```

用连接工具连接服务器后，在任意位置输入以上命令，按提示输入sigma用户名和密码即可。

2. 下载Shiny

```
$ sudo su - \  
-c "R -e \"install.packages('shiny', repos='https://cran.rstudio.com/')\""
```

通过以上命令可以安装Shiny程序包。也可以进入R程序安装R包，但注意如果装在private路径下，容易出问题。

3. 下载Shiny Server

```
$ sudo apt-get install gdebi-core  
$ wget https://download3.rstudio.org/ubuntu-12.04/x86_64/shiny-server-1.5.6.875-amd64.deb  
$ sudo gdebi shiny-server-1.5.6.875-amd64.deb
```

依次输入以上命令，安装Shiny Server。安装成功后，打开浏览器，输入 `10.53.0.3:3838`。注意如果是外网，需连接vpn；Shiny Server的默认端口是3838，当然后期可以更改。

shiny server启起来之后，前段会显示欢迎界面：

这个界面有3个程序组成，一个html程序，一个shiny程序，一个rmd程序。

* html程序的存储目录是：`/srv/shiny-server`；

* shiny程序的存储目录是 `/srv/shiny-server/sample-apps/hello` 下：`ui.R`, `server.R`;

* rmd程序的存储目录是 `/srv/shiny-server/sample-apps/rmd`：`index.Rmd`。

程序的逻辑是：html是网页，其中有两个iframe（可以简单理解为窗口或组件），两个iframe里放着shiny和rmd的可视化程序并可以在网页中展示可视化结果。

Shiny Server进阶版教程参见Administrator’s Guide: <http://docs.rstudio.com/shiny-server/>。

4. 部署Shiny代码踩过的坑

- 4.1 数据和代码要放到shiny server目录下

这里将数据和代码copy了一份放到 `/srv/shiny-server/sample-apps/hello` 路径下，这样shiny server才可以读到shiny程序依赖的数据和文件。

```
sigma@ubuntu:/srv/shiny-server/sample-apps$ cd hello
sigma@ubuntu:/srv/shiny-server/sample-apps/hello$ ls
README.md  fn  index of truck with Y unscale.csv  main_branch  main_test  score_truck_using_grouped_feature.csv  truck.cn.Rproj
data  index of truck with Y scaled.csv  main.R  main_data_generation  pic  server.R  ui.R
```

- 4.2 shiny server编译不过去不会报错

假设编译正确，但程序运行过程报错，会显示在前端。比如如下红字部分展示的错误：



但如果编译错误，前端不会有任何报错，只是需要的结果显示不出来。

部署过程最常见的编译错误，就是一个R脚本中，依赖的R包没有导入。该代码依赖的所有脚本和R包都要导入（source了一个R脚本，被source的脚本中导入了需要的R包也不行）。

如果想测试代码，可以用shiny用户登录服务器，在服务器上逐个通过 `Rscript` 命令跑通脚本。

- 4.3 脚本里的路径都要写对（这里省去一把辛酸泪）

在改脚本过程中，发现各种神奇问题。比如fn文件夹下有2个脚本，findTrip.R和accessAddress.R。而前者依赖于后者。在前者的脚本里，如下写着：

A terminal window with a black background and green text. The prompt is 'sigma@ubuntu:/srv/shiny-server/sample-apps/hello/fn\$'. The command 'cat findTrip.R' is entered. The output shows 'library(dplyr)' followed by 'source("fn/accessAddress.R")' which is highlighted with a red rectangle. Below it, '## user defined fn' is visible.

```
sigma@ubuntu:/srv/shiny-server/sample-apps/hello/fn$ cat findTrip.R
library(dplyr)
source("fn/accessAddress.R")
## user defined fn
```

这是错误的。

如果要写绝对路径，要写

全：`source("/srv/shiny-server/sample-apps/hello/fn/accessAddress.R")`；

如果要写相对路径，要写：`source("accessAddress.R")`，因为这两个脚本在同一目录下。像截图中那么写，程序找不到文件.....

在服务器上部署Shiny应用时，主要踩了以上坑，因为编译错误前端不报错等原因，几乎是把程序copy到本地，逐个脚本一行一行地运行，才跑通文件。是以为记，也做分享，争取以后不重复踩坑。（^__^）