Dealing with RSelenium

Aaron Rudkin

Installing RSelenium

First, install RSelenium

```
install.packages("RSelenium")
```

In the background, RSelenium will install `wdman` (web driver manager) and `binman` (binary manager)

Additionally, it will be useful to install the `netstat` package:

```
install.packages("netstat")
```

Attempt to run RSelenium

Attempt to run RSelenium as follows:

```
library(RSelenium)

rD <- rsDriver(
  browser = "chrome",
  check = TRUE,
  verbose = TRUE
)

rD$server$log()
```

Possible error 1: Java not installed

Selenium uses the Java Runtime Environment to be able to run its server. You must install the Java Runtime Environment. If you receive an error talking about Java, this should be the first thing you try to fix.

If you are on Windows, you can download it here (under Windows x64, you want the *.exe* version): https://www.oracle.com/java/technologies/javase-jre8-downloads.html

If you are on a Mac with an Intel processor, you can download from the same link (you want macOS x64 Installer): https://www.oracle.com/java/technologies/javase-jre8-downloads.html

If you are on a Mac with an M1 processor, you can download from this link (you want Java 8 (LTS) and .dmg file): https://www.azul.com/downloads/?os=macos&architecture=arm-64-bit&package=jre

If you are on a Mac and do not know which kind of processor you have, you can click the Apple icon in the top left hand corner, then "About this Mac". Observe whether the box mentions Intel or M1.

Possible error 2: Chrome version does not match chromedriver version

RSelenium uses software called chromedriver to talk to Chrome. Chrome and chromedriver must have the same version. By default, RSelenium will download the newest two versions of chromedriver. Sometimes this means that the newest version of chromedriver is designed for beta or unreleased versions of Chrome.

You can check your version of Chrome by opening Chrome, clicking the "hamburger" (three dots) menu on the top right side, help, About Google Chrome. Make a note of the "major" version number – the first number, typically 87, 88, 89, 90, or 91.

Now, you can check which versions of chromedriver you have by running this code:

```
binman::list_versions("chromedriver")
```

By default, RSelenium will use the highest version of chromedriver. If this is higher than your version of Chrome, you will have an error. You can solve this by using the `chromever` argument in the rsDriver command, i.e.:

```
rD <- rsDriver(
  browser = "chrome",
  # just an example, pick the major version that matches Chrome for you
  chromever = "90.0.4430.24",
  verbose = TRUE
)
```

Possible error 3: Port issues

Your computer listens for network signals using a system called "ports", which are numbers from 1 to 65,535; each program reserves one or more ports to listen for signals on. Lower numbers are reserved for core functionality, and so when running new programs they typically listen 4,000+. Selenium listens on 4567 by default.

A common occurrence is that your Selenium instance has an error, making it impossible for you to talk to it, but it continues to listen on the same port. Attempting to run Selenium again will cause an error because the port is already in use.

This can be solved by using the `port` argument and using the `netstat` package to ask Selenium to only listen on a port that is currently empty:

```
rD <- rsDriver(
  browser = "chrome",
  port = netstat::free_port(),
  verbose = TRUE
)
```

Possible error 4: Intel Mac downloads M1 Mac files instead

If you are running an Intel Mac, you may experience an issue where RSelenium accidentally downloads files intended for M1 Macs instead. We can fix this fairly easily. First, locate the chromedriver directory:

```
print(binman::app_dir("chromedriver"))
```

Copy this directory name. Open the Finder application. Click "Go", "Go To Folder", and paste the folder. Double click "mac64", then the correct chromedriver version (see above for discussion on chromedriver versions). You should see three files: chromedriver, chromedriver_mac64.zip, and chromedriver_mac64_m1.zip. Delete chromedriver. Double click chromedriver_mac64.zip, and a new chromedriver will appear. Finally, press and hold the "control" button and double-click chromedriver. If you are asked whether or not to open the file, select to open the file.

After this, Terminal.app will open. Click Terminal, Quit Terminal. We opened the file because macOS has default security intended to prevent running unknown files, and by manually opening it, we are giving it permission to run.

Annoyance: Downloading versions of drivers every time

By default, RSelenium tries to download every version of every browser driver every time you run it. This is not useful. You can solve this problem by using the `check = FALSE` argument.

Final code

This is the code I use to launch an RSelenium instance:

```
rD <- rsDriver(
  port = netstat::free_port(),
  browser = "chrome",
  chromever = "90.0.4430.24",
  check = FALSE
)
```

If RSelenium works, this command will open an empty version of Chrome which has the banner "Chrome is being controlled by automated test software."

Possible error 5: Unable to close Selenium

Closing Selenium is a little more difficult than you might imagine. These commands should work on Mac:

```
rD$client$closeall() # Closes the Chrome window
rD$server$stop() # Asks Selenium to stop nicely
rm(rD) # Removes the R connection to Selenium
gc() # Asks R to remove all memory associated with Selenium
```

On Windows, you may have to additionally use this code to shut down Selenium server:

```
system("taskkill /im java.exe /f", intern=FALSE, ignore.stdout=FALSE)
```