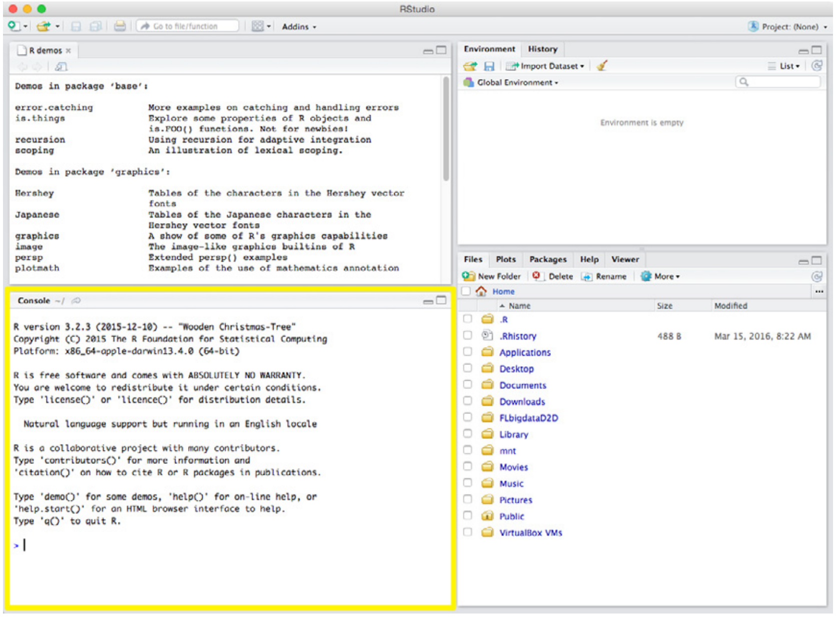
Instructions: Exploring data

In this exercise you will use RStudio and H2O to explore our banking dataset.

1) Open RStudio. Enter each command in the steps below, one line at a time, into your RStudio console.



```
Demos in package 'base':

error.catching          More examples on catching and handling errors
is.things               Explore some properties of R objects and
                        is.FOO() functions. Not for newbies!
recursion               Using recursion for adaptive integration
scoping                 An illustration of lexical scoping.

Demos in package 'graphics':

Hershey                 Tables of the characters in the Hershey vector
                        fonts
Japanese                Tables of the Japanese characters in the
                        Hershey vector fonts
graphics                A show of some of R's graphics capabilities
image                   The image-like graphics builtins of R
persp                   Extended persp() examples
plotmath                Examples of the use of mathematics annotation
```

```
R version 3.2.3 (2015-12-10) -- "Wooden Christmas-Tree"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

2) Load R packages. Each time we open RStudio we need to load our packages for that session.

```
library(dplyr)

library(ggplot2)

library(h2o)
```

3) Start the H2O server locally.

```
localH2o = h2o.init(ip = "127.0.0.1", port = 54321)
```

4) Import the dataset.

a) Load the file path into a variable:

```
filePath = "~/FLbigdataStats/bank_customer_data.csv"
```

b) Load the dataset and save it to the local handle 'market_data':

```
market_data <- h2o.uploadFile(filePath,
                              destination_frame = "",
                              parse = T,
                              header = T,
                              sep = ",",
                              na.strings = c("unknown"),
                              progressBar = FALSE,
                              parse_type = "CSV")
```

Note: In our exercises the H2O server is local so the data ends up in our RAM. If the server were in the cloud the data would be stored there.

5) Inspect the data.

a) Print a summary of the data frame, fetched from the H2O server:

```
market_data
```

b) Fetch summary statistics for columns from the server:

```
summary(market_data)
```

c) Inspecting big data in R is tricky. You don't want to load too much and exhaust your memory. We split the data into 20%, 80% slices and keep the 20% which is 8237 rows.

```
sample_frame <- h2o.splitFrame(market_data, ratio = 0.2)[[1]]
market_data_sample <- as.data.frame(sample_frame)
```

6) View the take-up by job.

   a) Let's have a look at offer take-up by job. This makes a table.

      ```
      by_y_job <- market_data_sample %>% group_by(y, job) %>%
      tally()
      ```

   b) View the table data.

      ```
      by_y_job
      ```

7) Plot the data.

   We can now plot the take-up by job with ggplot2.

   ```
   ggplot(data = by_y_job, aes(x = job, y = n, fill = y)) +
   geom_bar(stat = "identity", position = "dodge")
   ```