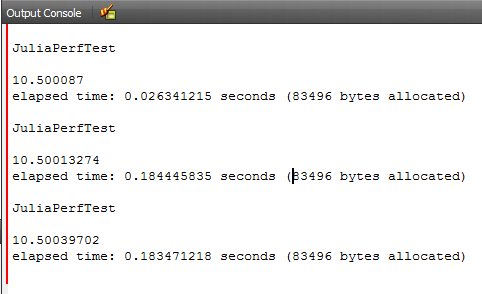
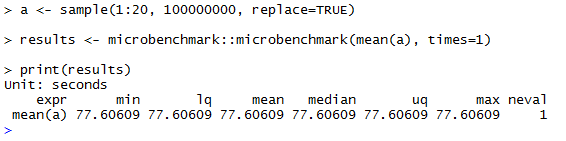
## 1. R vs Julia Performance Test

I chose to compare Julia vs R for a performance test of a simple mean calculation. The Julia code and a screenshot of the output are below, followed by the R code and associated screenshot.

|  |
| --- |
| Julia Code |
| function mean(a)  sum = 0  count = length(a)  for x in a  sum = sum + x  end  m = sum / count    println()  m  end  b = Array(Int32, 100000000)  rand!(1:20, b)  @time mean(b) |



| R Code |
| --- |
| require(microbenchmark)  mean <- function(a)  {  # Local Vars  sum <- 0  count <- length(a)    for ( x in a)  {  sum <- sum + x  }    m <- sum / count    # Return  return(m)  }  a <- sample(1:20, 100000000, replace=TRUE)  results <- microbenchmark::microbenchmark(mean(a), times=1)  print(results) |



### Results

Julia was significantly faster than R in the mean calculation. As shown, the code was written to be more similar rather than to code specifically for the advantages of either language (vectorized sum function in R for example). Julia took ~ 1/5 of a second, while R took ~77 seconds. In other words, R was ~421 times slower than Julia to perform this calculation!

## 2. Stored Procedure

The following PostgreSQL function (stored procedure) was written to insert or update a row in the ObjectList table and return the associated id to the caller.

|  |
| --- |
| CREATE OR REPLACE FUNCTION  upsert\_description(thename text, thedescription text)  RETURNS INT  AS $$  DECLARE theid INT;  BEGIN  SELECT ID FROM ObjectList WHERE name = theName INTO theid LIMIT 1;  IF theid IS NOT NULL THEN  UPDATE ObjectList SET description = theDescription WHERE ID = theid;  ELSE  INSERT INTO ObjectList (name, description)  VALUES (thename, thedescription) RETURNING id INTO theid;  END IF;  RETURN theid;  END;  $$  LANGUAGE plpgsql; |

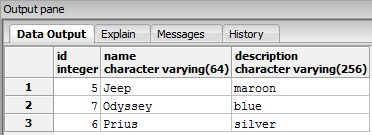
The ObjectList table was created using the following statement.

|  |
| --- |
| CREATE TABLE ObjectList  (  id serial PRIMARY KEY,  name VARCHAR(64),  description VARCHAR(256)  ); |

The following statements were used to insert and update data in the table, and finally query all the data for display.

|  |
| --- |
| SELECT upsert\_description('Jeep', 'maroon');  SELECT upsert\_description('Prius', 'gold');  SELECT upsert\_description('Odyssey', 'blue');  SELECT upsert\_description('Prius', 'silver');  SELECT \* FROM ObjectList; |

The following screenshot illustrates the contents of the ObjectList table after executing the statements listed above.



All code associated with this Quiz is available on [my GitHub](https://github.com/dwdii/DataAcqMgmt/tree/master/Week13).