• Share insights from engagements with customers in Finance who are exploring R
• Currently use SAS for all analytics work
• Shared a SAS script and asked us to replicate the work in R and Microsoft R
• Two small datasets were shared, with all variable names masked for security concerns
R should Provide Better Support for Dates

<table>
<thead>
<tr>
<th>CardID</th>
<th>CDate</th>
<th>ReportDate</th>
<th>month</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6/10/2015</td>
<td>9/1/2015</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>7/15/2015</td>
<td>9/1/2015</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>8/20/2015</td>
<td>9/1/2015</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>6/1/2015</td>
<td>9/1/2015</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>7/2/2015</td>
<td>9/1/2015</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>8/31/2015</td>
<td>9/1/2015</td>
<td>1</td>
</tr>
</tbody>
</table>

SAS

data bank;
set bank;
format CDate ReportDate date9. ;
month = intck('month', CDate, ReportDate);
run;

R

difftime(time1, time2, tz, units = c("auto", "secs", "mins", "hours", "days", "weeks"))

month_diff <- function(d1, d2) {
  d2_month  <- as.POSIXlt(d2)$year * 12 + as.POSIXlt(d2)$mon
  d1_month  <- as.POSIXlt(d1)$year * 12 + as.POSIXlt(d1)$mon
  d2_month  - d1_month
}

bank$month  <- month_diff(bank$CDate, bank$ReportDate)

MRS

my_func <- function(dataList){
dataList$month  <- month_diff(dataList$CDate, dataList$ReportDate)
return(dataList)
}

rollup_1 <- rxDataStep(inData = base_data_vara, outFile = rxXdfFile(tmpfile(tmpdir = temp_dir)), transformFunc = my_func, overwrite = TRUE)
SAS to R: Knowledge in Both is Necessary

- This example illustrates the mistakes one can make if there’s lack of knowledge in both
Deep Dive into the SAS Code

```sas
proc sort data = bank ;
  by CardID ;
run ;

data bank_mod (drop = i month Expense);
  set bank;
  by CardID;
  if n = 1 then do;
    array Expensex(3) Expense1 - Expense3;
  end;
  retain Expense1 - Expense3;
  if first.CardID then do;
    do i = 1 to 3;
      Expensex(i) = .;
    end;
  end;
  Expensex(month) = Expense;
  if last.CardID then output;
run ;
```

Block A

Block B

Block C

Block D

Block E
Key Observations from this Example

• This is a straightforward transformation in R and MRS
• SAS processes data row by row, allowing it to work with data of all sizes
• R processes all rows at the same time, making it subject to the memory size limit
• Microsoft R allows us to overcome R’s limits on data size
• Business knowledge is helpful for understanding the logic

R

```r
bank_mod <- reshape(bank,
                    v.names = "Expense",
                    idvar = "CardID",
                    timesvar = "month",
                    direction = "wide")
```

```r
require(data.table)
bank_mod <- dcast(bank,
                  CardID ~ month,
                  value.var = "Expense")
```

MRS

```r
library(xdfUtils)
outFile <- RxxDFData(tempfile(tmpdir = temp_dir))
```

```r
mydt_cast <- dcastXdf(inFile = inputFileNameXdf,
                      outFile = outFile,
                      CardID ~ month,
                      value.var = "Expense",
                      verbose = TRUE)
```
A Foreign Bank

• Currently use SAS
• Wanted to know how certain SAS operations can be done in R and Microsoft R
• Wanted to compare the speed between SAS and R/Microsoft R
Sample Functions the Customer Looked into

- SAS “first.” – multiple R options depending on context
- SAS “last.” – multiple R options depending on context
- SAS “compress” – “gsub” in R
- SAS “trim” – “trimws” in R
- SAS “intnx” – functions from the “lubridate” package
```r
transform_func <- function(dataList){
  # SAS compress: dropping blanks
dataList$customer1 <- gsub(" ", ",", dataList$customer)

  # SAS compress: dropping letter "a" and space " 
)dataList$customer2 <- gsub("[a ]", ",", dataList$customer)

  # SAS trim: removing trailing blanks
dataList$customer3 <- trimws(dataList$customer, which = "right")

  # SAS intnx(): add month
dataList$eventdate_month <- ymd(dataList$eventdate) +%m% months(1)

  # SAS intnx(): add weeks
dataList$eventdate_weeks <- ymd(dataList$eventdate) + weeks(2)

  # SAS intnx(): add days
dataList$eventdate_days <- ymd(dataList$eventdate) + days(29)

  return(dataList)
}

mydata_xdf_new <- rxDataStep(inData = mydata_xdf,
outFile = RxXdfData(tempfile(tmpdir = temp_dir)),
transformPackages = c("lubridate"),
transformFunc = transform_func,
overwrite = TRUE)
```
Operationalization - Tasks

**DOCUMENT MODELS**
Describe how calculations should be carried out

**ANSWER QUESTIONS**
Help IT understand the requirements

**TEST DEPLOYMENT**
Make sure model is deployed correctly
Operationalization - Challenges

Time Consuming
Both modelers and IT teams need to spend lots of time making sure they understand each other.

Error-Prone
Need double-check from both modelers and IT teams.

Complex Models
Difficult to implement complex models such as gradient boosting machines.
Operationalization – Opportunities

Make it Easy to Publish Models
Allow data scientists to publish models from R with a few lines of code

Support Different Consumption Modes
RESTful APIs can be accessed by any programming language, such as .NET, C#, Java, Javascript, Python, or node.js.

Allow Easy Update
Models can be updated from R with a few lines of code
Operationalization on Microsoft R Server

```r
# load library
library(mrsdeploy)
remoteLogin(server, username, password)

# publish a web service
publishService(service_name, code, model, inputs, outputs)

# call an existing web service
myapi <- getService(service_name)
result <- myapi$arrDelay(data)

# update a web service
updateService(service_name, code, model)
```
Why Microsoft Big Data & Advanced Analytics?
What Customers Tend to Care About

Functions
- Can R support various types of data manipulations
- Does R offer different modeling options

Scalability and Portability
- Does R work with data of all sizes, with speed
- We wrote Microsoft R code that runs both locally and on Spark: rxSetComputeContext()

Cost
- SAS is very expensive
- R and Microsoft R can be better alternatives
Resources

• Introduction to Microsoft R Server

• Microsoft R Documentation

• SAS to R Migration Blog Post