



Science For A Better Life



SAS, Java & Excel

A ménage à trois

PhUSE Annual Conference / October 2016 / Dana Vinzelberg



Science For A Better Life



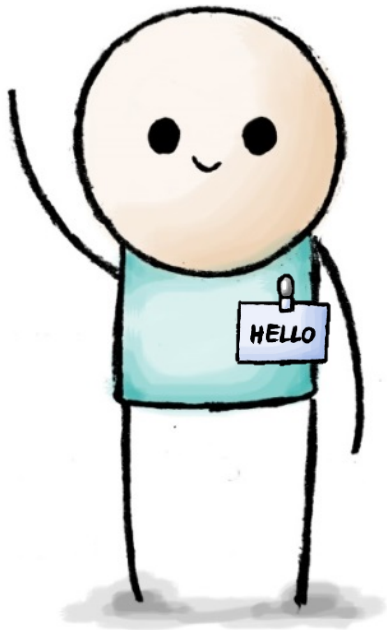
SAS, Java & Excel

A ménage à trois

PhUSE Annual Conference / October 2016 / Dana Vinzelberg

Outline

Introduction



Introduction



Communication



Obstacles



Improvement



Conclusion

Excel

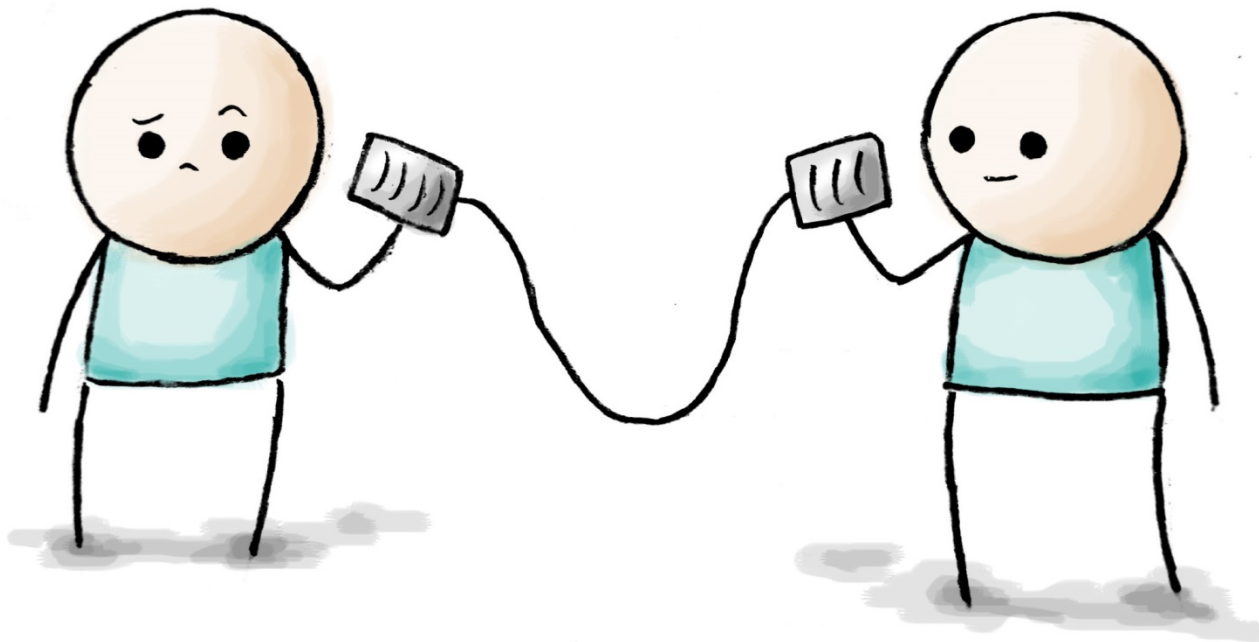


- Excel commonly known
- for data exchange
- Excel data accessible in SAS
- process and output



Communication

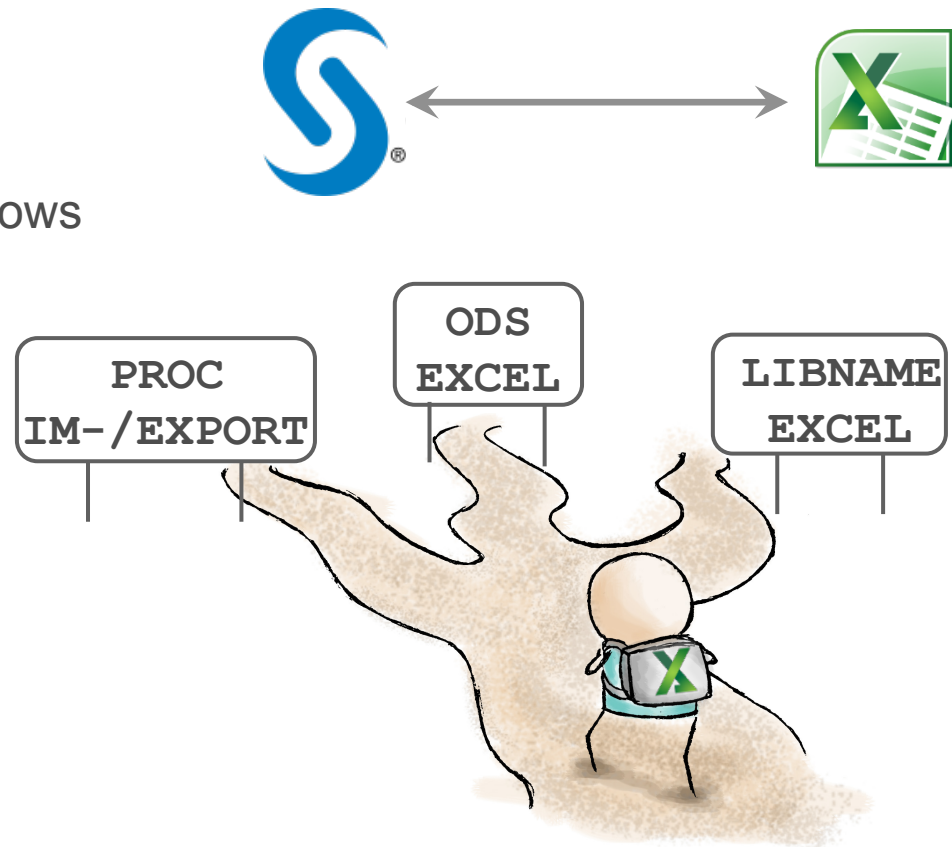
Introduction Communication





SAS and Excel

- many different ways
- PROC IM-/EXPORT
 - DBMS=EXCEL only for Windows
 - DBMS=XLS differently
- ODS EXCEL
 - not on native Excel format
- LIBNAME EXCEL
 - since SAS 9.3
 - lock on Excel file whilst active





Excel and Java

- intermediate step: Java
- many free packages
- Apache POI HSSF/XSSF
- Access to Excel file





Java and SAS

- utilize Apache POI HSSF/XSSF
- Java class for reading Excel
- pack into JAR (Java ARchive)



```
class workbook
```

```
constructor(Excel-file)
```

```
String get_cell_value(col,row)
```



%Excel2SAS

- utilize Apache POI HSSF/XSSF
- Java class for reading Excel
- pack into JAR (Java ARchive)
- accessible in SAS
- JavaObj statement in Data Step
- create Dataset(s)
- Import Excel data to SAS



```
class workbook
```

```
constructor (Excel-file)
```

```
String get_cell_value(col,row)
```

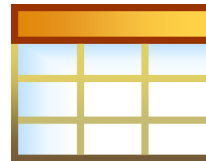
```
%MACRO ();  
  %add_to_CLASSPATH;  
  ...  
  DATA;  
    ATTRIB passing_value LENGTH=$200;  
    DECLARE JavaObj wb ('workbook', Excel);  
    wb.callStringMethod('get_cell_value',  
                        2, 4,  
                        passing_value);  
  
  RUN;  
  ...  
%MEND;
```



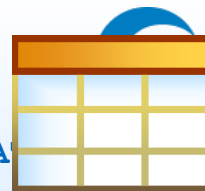


%SAS2Excel

- Export Dataset(s) to Excel
- get Dataset
- utilize Apache for Writing
- create Excel file in Java
- pass Dataset values



```
%MACRO ();  
...  
%add_to_CLASSPA  
...  
DATA;  
    DECLARE JavaObj();  
RUN;  
...  
%MEND;
```



Obstacles

Communication



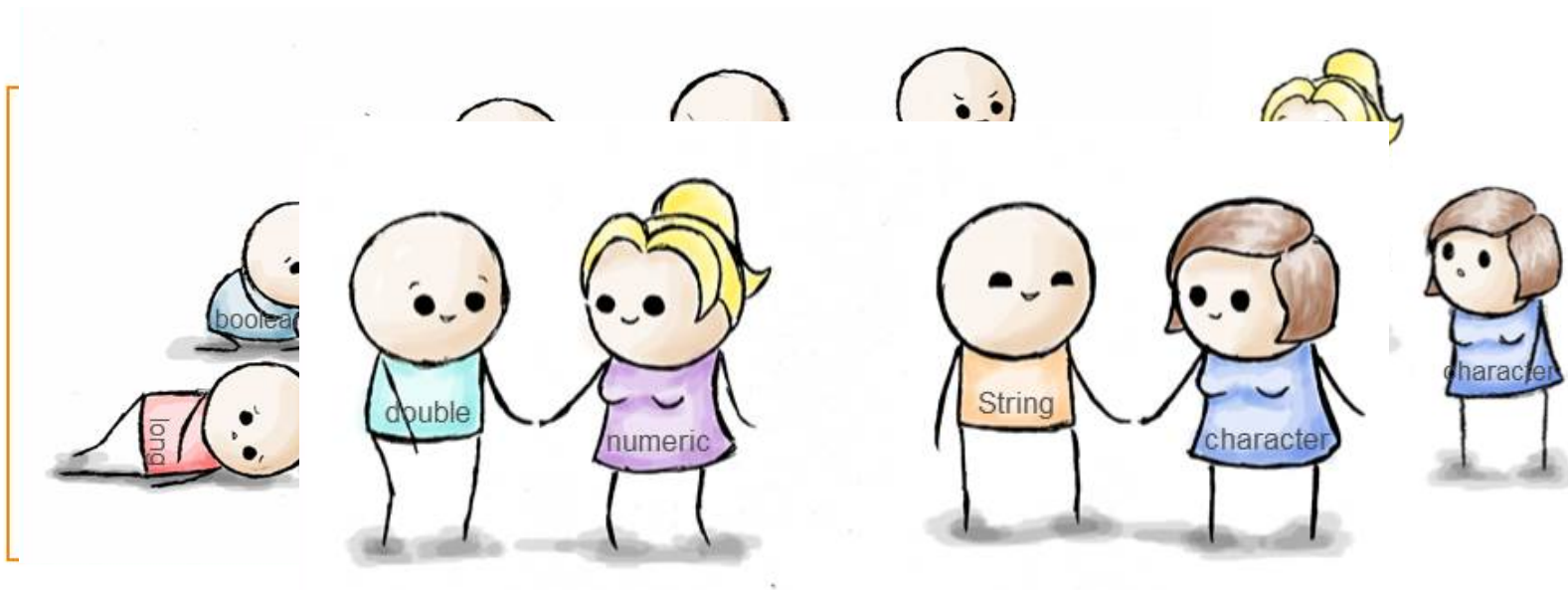
Obsta





Data Types

- different amount of data types: 9 vs. 2
- only two of them can match
- Java interprets numeric as double and character as string
- Java methods for communication with SAS:





Data Types

- other way around
- SAS functions for each Java data type

TYPE	JAVA OBJ METHODS
void	call Void Method
double	call Double Method
String	call String Method
char	call Char Method
Boolean	call Boolean Method
short	call Short Method
byte	call Byte Method
long	call Long Method
float	call Float Method
int	call Int Method

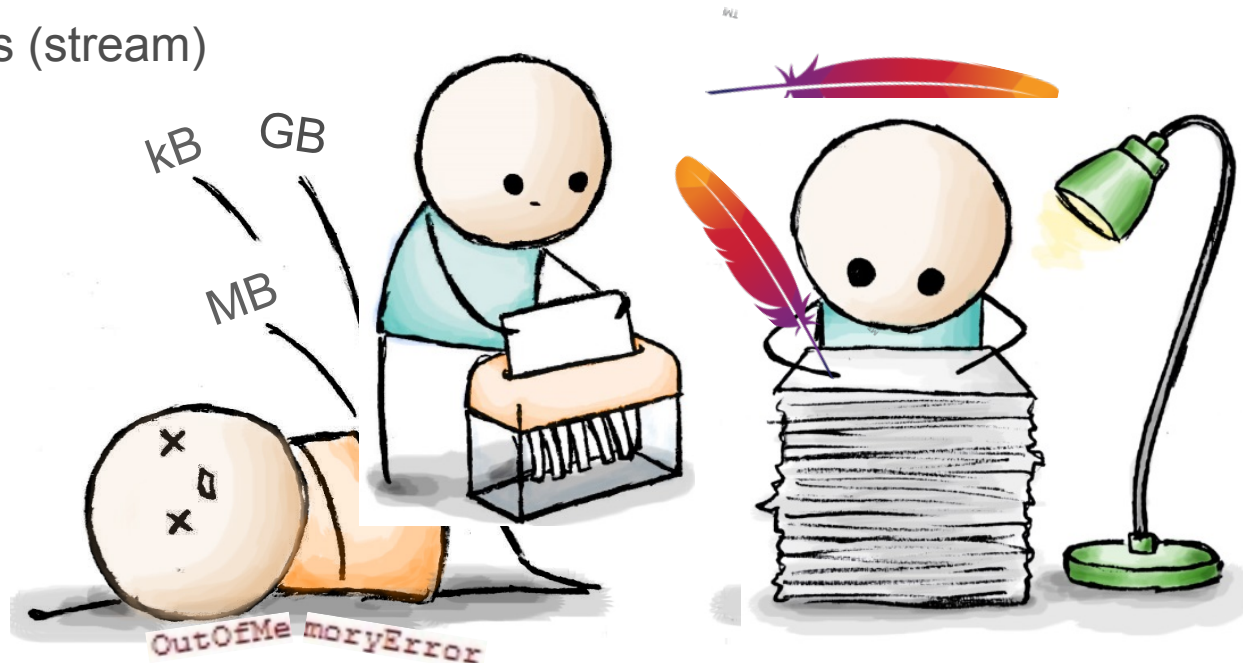
Large Files

Writing (aka The Fight with the Heap Space)

Obstacles



- problem with large datasets
- `java.lang.OutOfMemoryError: Java heap space`
- Java working space = Heap Space
- Apache provides SXSSF
- Writing in parts (stream)

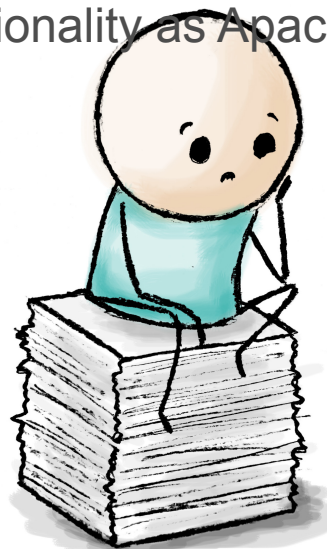


Large Files Reading

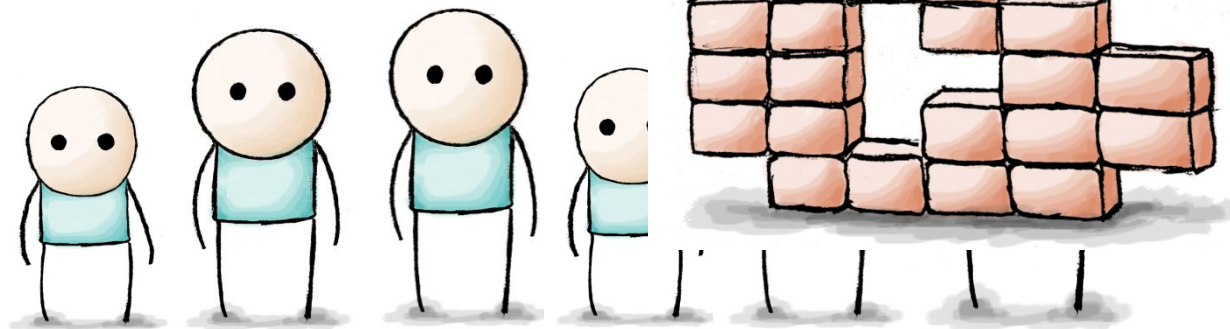
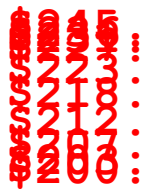
Obstacles



- Apache provides nothing for faster reading
- reading with SAX parser in Java
- not on native Excel file but XML version
- adapted to provide same functionality as Apache



- unevenly filled tables
 - filled up with empty cells
- format length



Improvement

Obstacles



Improvement

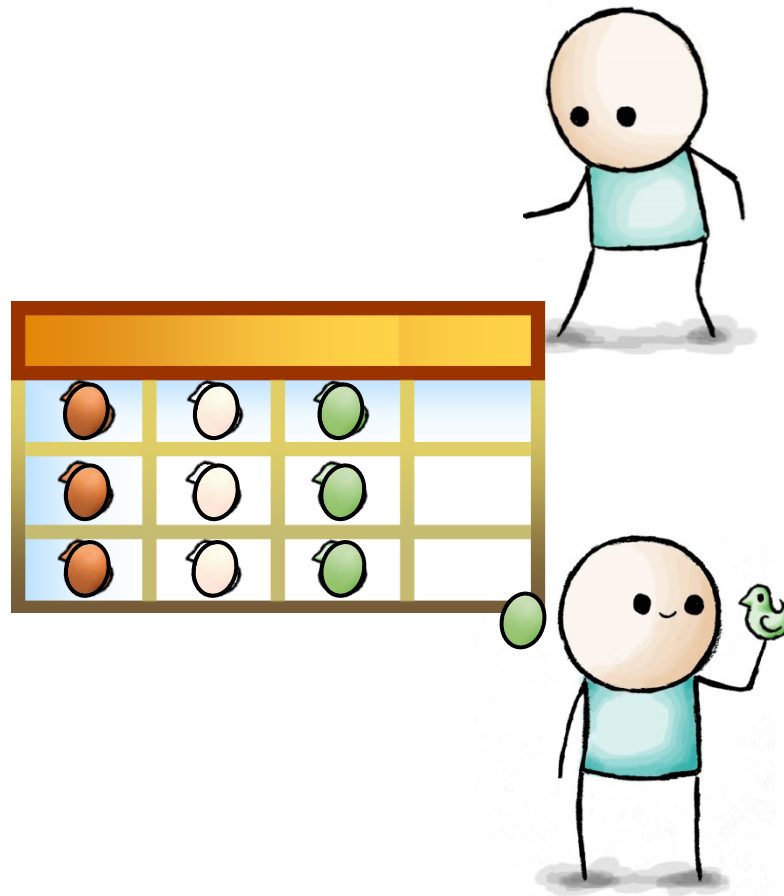


Enhanced Functionality

Improvement



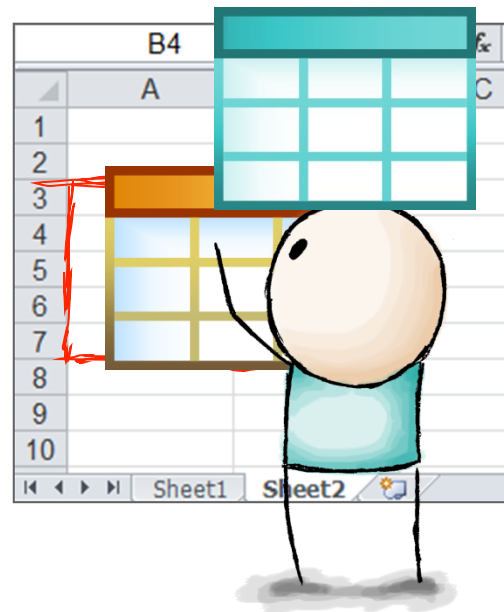
- through Parameter
- %SAS2Excel
 - exclude variables
 - raw values
or formatted values
- %Excel2SAS





Enhanced Functionality

- through Parameter
- %SAS2Excel
 - exclude variables
 - raw values
 - or formatted values
- %Excel2SAS
 - sheet range
 - reference dataset





The Power of Three

- SAS: statistics and big data
- Java: platform independence
- Excel: commonly known
- all advantages together in two macros:
 - %SAS2Excel
 - %Excel2SAS





Conclusion

- ménage à trois $\Rightarrow$ great tools
- implementation easy for SAS programmers
 - with interest in Java



Conclusion



Questions?

Conclusion



Thank You!