



# *Let's get Groovy - Word and PDF processing in SAS*



November 2022  
Katja Glaß





# Agenda

- // Introduction
- // Java Options
- // PROC GROOVY
- // PDF Processing
- // Docx Processing
- // Summary

# Introduction

- SAS very powerful for data processing
  - Mapping
  - Derivations
  - Statistics
  - Output generation



Document  
Processing



Web  
Generation

Designs



# Introduction

➤ SAS & Java



➤ Read in PDFs

➤ Read tables from Word



# *Java Options*



# Java Options

## ➤ Java via Command Line

Command Line

X 'java helloWorld';

```
CHL Eingabeaufforderung
Microsoft Windows [Version 10.0.19042.2006]
(c) Microsoft Corporation. Alle Rechte vorbehalten.
H:\>java HelloWorld_
```



# Java Options

## ➤ JavaObj

### JavaObj

```
DATA _NULL_;  
  DCL JAVAOBJ j("helloWorldClass");  
  rc = j.callStringMethod("getHelloWorld");  
  PUT rc=;  
RUN;
```



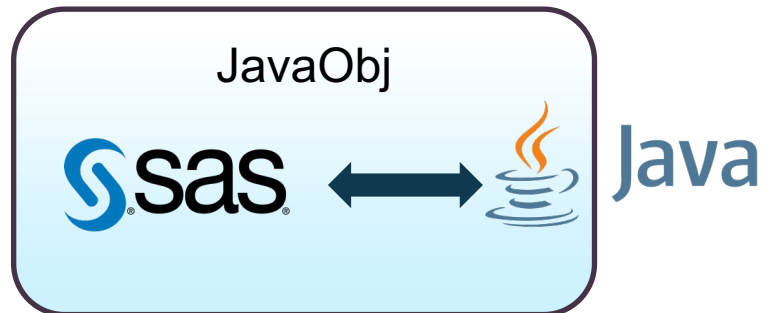
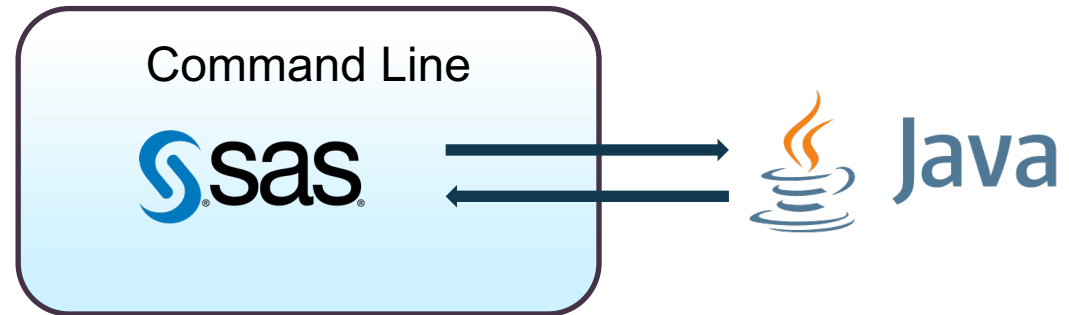
# Java Options

## ➤ PROC GROOVY

### PROC GROOVY

```
PROC GROOVY;  
  SUBMIT;  
    System.out.println("Hello world!");  
  ENDSUBMIT;  
QUIT;
```

# Java Options





# *PROC GROOVY*

# PROC GROOVY

## Syntax

```
PROC GROOVY;  
    ADD CLASSPATH = "<path>/<file>.jar";  
    SUBMIT;  
        <Java source>  
    ENDSUBMIT;  
QUIT;
```





# PROC GROOVY

## Syntax

```
%LET name =  
PROC GROOVY  
    SUBMIT;  
        SysMacro Hello (&name!) "  
    ENDSUBM  
QUIT;
```

Hello &name!





# PROC GROOVY

## Syntax

```
%LET name = KATJA;  
PROC GROOVY;  
    SUBMIT "&  
        String  
        System  
    ENDSUBMIT  
QUIT;
```

Hello Katja!

```
+ name + "!" );
```



# PROC GROOVY

## Syntax

```
%MACRO sayHello(name);
```

```
    PROC GROOVY;
```

```
        SU
```

```
        St
```

```
        Sy
```

```
    ENDSU
```

```
    QUIT;
```

```
%MEND;
```

```
%sayHello(Katja);
```

**ERROR: Submit block cannot be directly placed in a macro. ...**



```
name + "!" );
```



# PROC GROOVY

## Syntax

```
%MACRO sayHello(name);  
    FILENAME cmdFile TEMP;  
    DATA _NULL_;  
        FILE cmdFile LRECL=500;  
        PUT 'PROC GROOVY;';  
        PUT '    ...';  
    RUN;  
    %INCLUDE cmdFile;  
%MEND sayHello;  
%sayHello(Katja);
```





# *PDF Processing*



# PDF Processing

Read PDF into SAS

- Working with PDF Documents
  - Extract content (Protocol, Deliverables, annotated CRF)
  - Compare PDF for equal content
  - ...
  
- Options with Java





# PDF Processing

## Read PDF into SAS

```
import ...;
public class ReadingText {
    public static void main(String args[]) throws IOException {
        //Loading an existing document
        File file = new File("C:/PdfBox_Examples/new.pdf");
        PDDocument document = PDDocument.load(file);

        //Instantiate PDFTextStripper class
        PDFTextStripper pdfStripper = new PDFTextStripper();

        //Retrieving text from PDF document
        String text = pdfStripper.getText(document);
        System.out.println(text);

        //Closing the document
        document.close();
    }
}
```



[https://www.tutorialspoint.com/pdfbox/pdfbox\\_reading\\_text.htm](https://www.tutorialspoint.com/pdfbox/pdfbox_reading_text.htm)



# PDF Processing

## Read PDF into SAS

```
import ...;
public class ReadingText {
    public
        import ...;
        public class ReadingText {
            public static void main(String args[])
                throws IOException {
                ...
            }
        }
    }
}
```



# PDF Processing

## Read PDF into SAS

```
import ...;
public class ReadingText {

    File file = new File("C:/PdfBox_Examples/new.pdf");
    PDDocument document = PDDocument.load(file);
    PDFTextStripper pdfStripper = new PDFTextStripper();
    String text = pdfStripper.getText(document);

}
}
```



# PDF Processing

## Read PDF into SAS

```
import ...;
public class ReadingText {
    public static void main(String args[]) throws IOException {
        //Loading an existing document
        File file = new File("C:/PdfBox Examples/new.pdf");
        PDDocument document = new PDDocument(file);

        //Instantiating a PDFTextStripper
        PDFTextStripper stripper = new PDFTextStripper();

        //Retrieving the text
        String text = stripper.getText(document);
        System.out.println(text);

        //Closing the document
        document.close();
    }
}
```

```
System.out.println(text);
document.close();
```



# PDF Processing

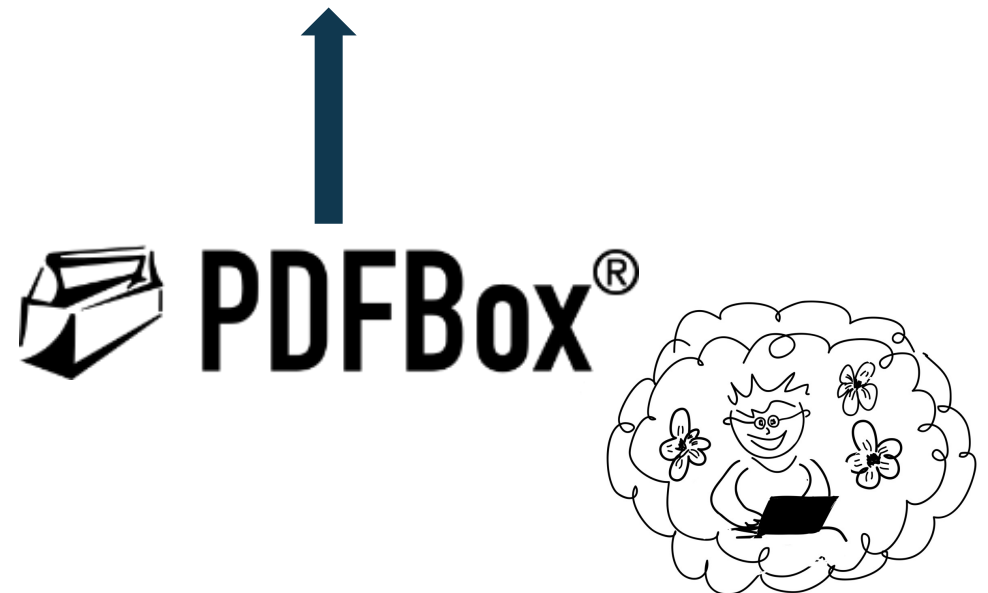
Read PDF into SAS



# PDF Processing

## Read PDF into SAS

```
%LET infile = <path>/sdtmadampilotprojectreport.pdf;  
%LET pdfboxPath = <path>;  
PROC GROOVY;  
    ADD CLASSPATH = "&pdfboxPath./pdfbox-app-2.0.26.jar";  
    SUBMIT "&infile";  
    ...  
ENDSUBMIT;  
QUIT;
```





# PDF Processing

Read PDF into SAS

```
String infile = args[0];
```

```
import org.apache.pdfbox.pdmodel.PDDocument;  
import org.apache.pdfbox.text.PDFTextStripper;
```

No import of  
„standard java“ required





# PDF Processing

Read PDF into SAS

Copy / Rewrite Code

```
// load document
```

```
File
```

```
PDDoc
```

```
24          //Closing the document  
25          document.close();
```

```
// ge
```

```
PDFTe
```

```
26          ENDSUBMIT;  
CDISC SDTM/ADaM Pilot Project1  
Project Report  
Executive Summary
```

```
// ge
```

```
Strin
```

```
Syste
```

```
Background  
CDISC is a non-profit, multidisciplinary consensus based st  
organization founded over a decade ago that has established
```

```
// closing document  
document.close();
```





# PDF Processing

Read PDF into SAS

Save to TXT

```
// Write text to file  
FileWriter myWriter = new FileWriter(outtxt);  
myWriter.write(text);  
myWriter.close();
```





# PDF Processing

Read PDF into SAS





# *DOCX Processing*



# DOCX Processing

## Tables to SAS

- Working with DOCX Documents
  - Extract content (Protocol, Deliverables, annotated CRF)
  - Compare DOCX for equal content
  - Investigate tables
- Options with Java





# DOCX Processing

Tables to SAS



## Apache POI files

| Name                     |
|--------------------------|
| auxiliary                |
| lib                      |
| ooxml-lib                |
| LICENSE                  |
| NOTICE                   |
| poi-5.2.3.jar            |
| poi-examples-5.2.3.jar   |
| poi-excelant-5.2.3.jar   |
| poi-javadoc-5.2.3.jar    |
| poi-ooxml-5.2.3.jar      |
| poi-ooxml-full-5.2.3.jar |
| poi-ooxml-lite-5.2.3.jar |
| poi-scratchpad-5.2.3.jar |



## Apache POI bundle

| Name                                          |
|-----------------------------------------------|
| org.apache.servicemix.bundles.poi-5.2.2_3.jar |



## Apache POI final files

| Name                                          |
|-----------------------------------------------|
| commons-compress-1.21.jar                     |
| org.apache.servicemix.bundles.poi-5.2.2_3.jar |

# DOCX Processing

## Tables to SAS

- Search the internet
  - Not perfect solution
  - Use as base, enhance peace-by-peace



<https://mkyong.com/java/java-read-and-write-microsoft-word-with-apache-poi/>

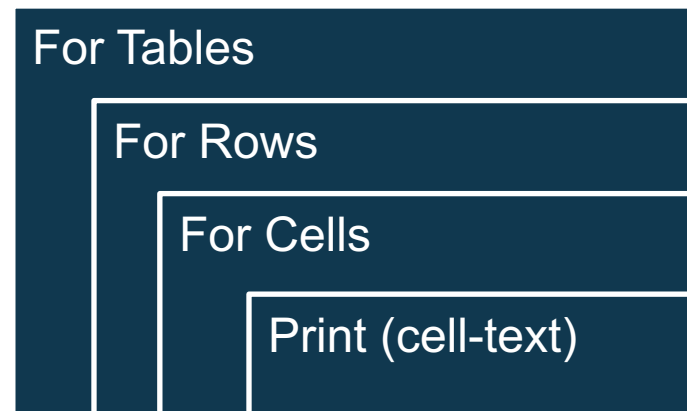




# DOCX Processing

## Tables to SAS

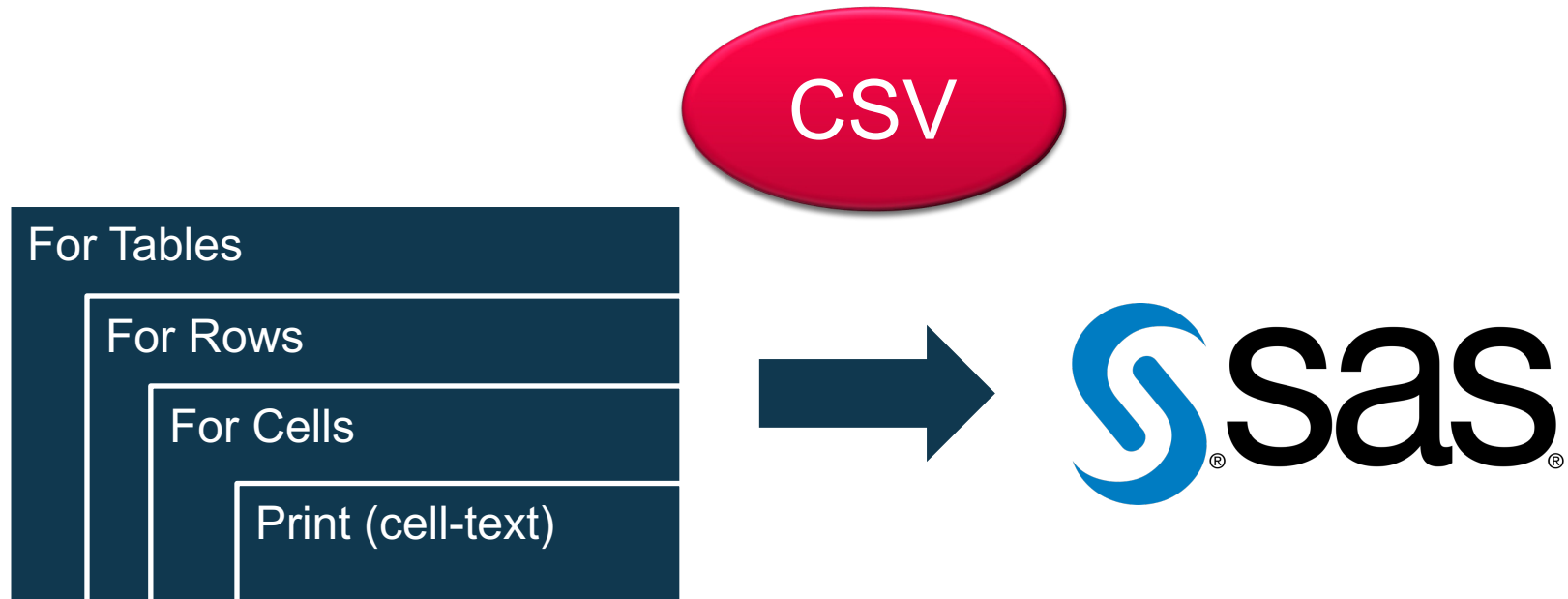
```
XWPFDocument doc = new
XWPFDocument(Files.newInputStream(Paths.get(fileName)))
List<XWPFTable> xwpfTableList = doc.getTables();
for (XWPFTable xwpfTable : xwpfTableList) {
    System.out.println("Total Rows : " + xwpfTable.getNumberOfRows());
    for (int i = 0; i < xwpfTable.getRows().size(); i++) {
        for (int j = 0; j < xwpfTable.getRow(i).getTableCells().size(); j++) {
            System.out.println(xwpfTable.getRow(i).getCell(j).getText());
        }
    }
}
```





# DOCX Processing

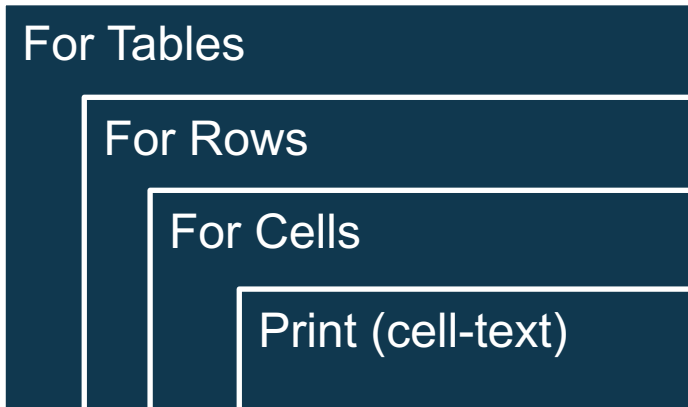
Tables to SAS



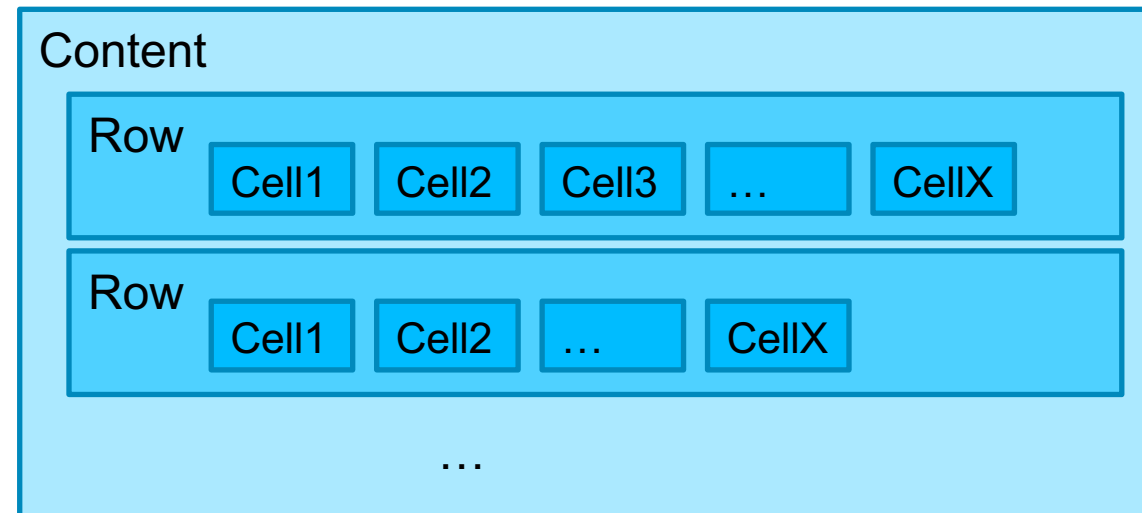


# DOCX Processing

## Tables to SAS



List<List<String>>





# PDF Processing

## Read PDF into SAS

### Initialization

```
%LET infile = <path>/14-2.01.docx;  
%LET outfile = %SYSFUNC(PATHNAME(work))/content_xx.csv;  
%LET jarPath = <path>;  
PROC GROOVY;  
    ADD CLASSPATH = "&jarPath./commons-compress-1.21.jar";  
    ADD CLASSPATH = "&jarPath./org.apache.servicemix.bundles.poi-5.2.2_3.jar";  
    SUBMIT "&infile" "&outfile";  
  
import org.apache.poi.xwpf.usermodel.XWPFDocument;  
import org.apache.poi.xwpf.usermodel.XWPFTable;  
import java.nio.file.Files;  
import java.nio.file.Paths;  
  
String inFile = args[0];  
String outFile = args[1];
```



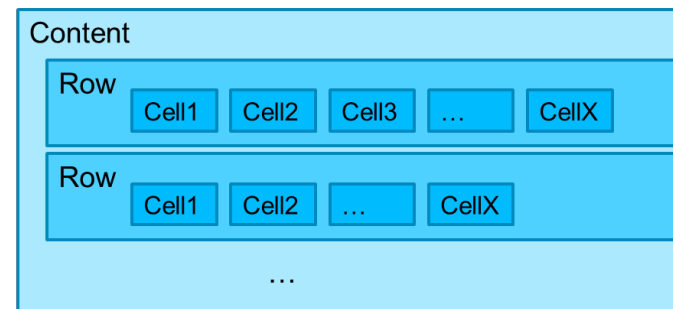


# PDF Processing

Read PDF into SAS

Store Cell Information

```
XWPFDocument doc = new XWPFDocument(Files.newInputStream(Paths.get(inFile)))
List<XWPFTable> xwpfTableList = doc.getTables();
List<List<String>> listOfLists = new ArrayList<>();
for (XWPFTable xwpfTable : xwpfTableList) {
    for (int i = 0; i < xwpfTable.getRows().size(); i++) {
        List<String> currentRow = new ArrayList<String>();
        for (int j = 0; j < xwpfTable.getRow(i).getTableCells().size(); j++) {
            currentRow.add(xwpfTable.getRow(i).getCell(j).getText());
        }
        listOfLists.add(currentRow);
    }
}
```



# PDF Processing

Read PDF into SAS

## CSV Export

```
FileWriter myWriter = new FileWriter(outFile);  
for (List<String> currentRow : listOfLists) {  
    myWriter.write(String.join("\t", currentRow));  
    myWriter.write('\n');  
}  
myWriter.close();
```

|  |  |  |
|--|--|--|
|  |  |  |
|  |  |  |
|  |  |  |



# PDF Processing

## Read PDF into SAS

Protocol: CDISCPiLOT01  
Population: Intent-to-Treat

### Summary

|         |           |
|---------|-----------|
| Age (y) | n         |
|         | Mean      |
|         | SD        |
|         | Median    |
|         | Min       |
|         | Max       |
|         | <65 yrs   |
|         | 65-80 yrs |
|         | >80 yrs   |
| Sex     | n         |
|         | Male      |
|         | Female    |

|    | VAR1    | VAR2      | VAR3      | VAR4      | VAR5      | VAR6       | VAR7   |
|----|---------|-----------|-----------|-----------|-----------|------------|--------|
| 1  | Age (y) | n         | 86        | 84        | 84        | 254        | 0.5934 |
| 2  |         | Mean      | 75.2      | 75.7      | 74.4      | 75.1       | .      |
| 3  |         | SD        | 8.59      | 8.29      | 7.89      | 8.25       | .      |
| 4  |         | Median    | 76.0      | 77.5      | 76.0      | 77.0       | .      |
| 5  |         | Min       | 52.0      | 51.0      | 56.0      | 51.0       | .      |
| 6  |         | Max       | 89.0      | 88.0      | 88.0      | 89.0       | .      |
| 7  |         |           |           |           |           |            | .      |
| 8  |         | <65 yrs   | 14 ( 16%) | 8 ( 10%)  | 11 ( 13%) | 33 ( 13%)  | 0.1439 |
| 9  |         | 65-80 yrs | 42 ( 49%) | 47 ( 56%) | 55 ( 65%) | 144 ( 57%) | .      |
| 10 |         | >80 yrs   | 30 ( 35%) | 29 ( 35%) | 18 ( 21%) | 77 ( 30%)  | .      |
| 11 |         |           |           |           |           |            | .      |
| 12 | Sex     | n         | 86        | 84        | 84        | 254        | 0.1409 |
| 13 |         | Male      | 33 ( 38%) | 34 ( 40%) | 44 ( 52%) | 111 ( 44%) | .      |
| 14 |         | Female    | 53 ( 62%) | 50 ( 60%) | 40 ( 48%) | 143 ( 56%) | .      |

|     |           |           |           |           |
|-----|-----------|-----------|-----------|-----------|
|     | <65 yrs   | 14 ( 16%) | 8 ( 10%)  | 11 ( 13%) |
|     | 65-80 yrs | 42 ( 49%) | 47 ( 56%) | 55 ( 65%) |
|     | >80 yrs   | 30 ( 35%) | 29 ( 35%) | 18 ( 21%) |
| Sex | n         | 86        | 84        | 84        |
|     | Male      | 33 ( 38%) | 34 ( 40%) | 44 ( 52%) |
|     | Female    | 53 ( 62%) | 50 ( 60%) | 40 ( 48%) |





# PDF Processing

Read PDF into SAS



|    | VAR1    | VAR2      | VAR3      | VAR4      | VAR5      | VAR6       | VAR7   |
|----|---------|-----------|-----------|-----------|-----------|------------|--------|
| 1  | Age (y) | n         | 86        | 84        | 84        | 254        | 0.5934 |
| 2  |         | Mean      | 75.2      | 75.7      | 74.4      | 75.1       | .      |
| 3  |         | SD        | 8.59      | 8.29      | 7.89      | 8.25       | .      |
| 4  |         | Median    | 76.0      | 77.5      | 76.0      | 77.0       | .      |
| 5  |         | Min       | 52.0      | 51.0      | 56.0      | 51.0       | .      |
| 6  |         | Max       | 89.0      | 88.0      | 88.0      | 89.0       | .      |
| 7  |         |           |           |           |           |            | .      |
| 8  |         | <65 yrs   | 14 ( 16%) | 8 ( 10%)  | 11 ( 13%) | 33 ( 13%)  | 0.1439 |
| 9  |         | 65-80 yrs | 42 ( 49%) | 47 ( 56%) | 55 ( 65%) | 144 ( 57%) | .      |
| 10 |         | >80 yrs   | 30 ( 35%) | 29 ( 35%) | 18 ( 21%) | 77 ( 30%)  | .      |
| 11 |         |           |           |           |           |            | .      |
| 12 | Sex     | n         | 86        | 84        | 84        | 254        | 0.1409 |
| 13 |         | Male      | 33 ( 38%) | 34 ( 40%) | 44 ( 52%) | 111 ( 44%) | .      |
| 14 |         | Female    | 53 ( 62%) | 50 ( 60%) | 40 ( 48%) | 143 ( 56%) | .      |



# *Summary*

# Summary

- SAS integration for Java very well
- Many examples for various tasks online
- Easy learn and adopt possible





*Thank you!*



**Bye-Bye**

