

Achieving a seamless workflow between R, Python, and SAS from within RStudio

`rstudio::conf(2022)`

Melissa Van Bussel

Statistics Canada

July 28th, 2022

Achieving a seamless workflow between R, Python, and SAS from within RStudio

Achieving a seamless workflow between R, Python, and SAS from within RStudio







Breakfast: Bunny Rabbit Pancakes

with fresh fruit and maple syrup



Preparation time: 25 mins

Cooking difficulty: Easy

Allergens: Egg • Wheat • Gelatin

Ingredients



1 unit
Pancake mix



0.6 lbs
Cantaloupe



1 unit
Egg



8 oz
Strawberries



1 tbsp
Vegetable oil



3oz
Blueberries



1 fl oz
Maple syrup



1 package
Marshmallows

Instructions

1

Combine egg, oil, and pancake mix.
Stir until large lumps disappear.

2

Lightly grease and heat skillet over medium-low heat. Pour batter into skillet for each pancake and turn when bottoms are golden brown.

3

Add cantaloupe slices for the ears, blueberries for the eyes, a strawberry for the nose, and marshmallows for the teeth. Drizzle with maple syrup as desired.





Breakfast: Bunny Rabbit Pancakes

with fresh fruit and maple syrup

Allergens: Egg • Wheat • Gelatin

Ingredients



1 unit
Pillsbury pancake mix



0.6 lbs
Cantaloupe



1 unit
Egg



8 oz
Strawberries



1 tbsp
Canola oil



3oz
Blueberries



1 fl oz
Table syrup



1 package
Marshmallows



Preparation time: 25 mins

Cooking difficulty: Easy

Instructions

1

Combine egg, oil, and pancake mix.
Stir until large lumps disappear.

2

Lightly grease and heat skillet over medium-low heat. Pour batter into skillet for each pancake and turn when bottoms are golden brown.

3

Add cantaloupe slices for the ears,
blueberries for the eyes, a
strawberry for the nose, and
marshmallows for the teeth. Drizzle
with maple syrup as desired.



Outline

I'll be discussing how to incorporate all three languages into your workflow

Outline

I'll be discussing how to incorporate all three languages into your workflow

- My ideal approach
 - 3 step recipe with all ingredients available

Outline

I'll be discussing how to incorporate all three languages into your workflow

- My ideal approach
 - 3 step recipe with all ingredients available
- My less ideal approach
 - A few ingredient substitutions

Who am I?

- Melissa Van Bussel, M. Sc. Statistics, A. Stat., B. Sc. Math & CS

Who am I?

- Melissa Van Bussel, M. Sc. Statistics, A. Stat., B. Sc. Math & CS
- Analyst at Statistics Canada (Government of Canada)



Statistics
Canada

Statistique
Canada



Who am I?

- Melissa Van Bussel, M. Sc. Statistics, A. Stat., B. Sc. Math & CS
- Analyst at Statistics Canada (Government of Canada)
- Statistics Canada = Canada's official statistical agency (over **6,000** employees)



Statistics
Canada

Statistique
Canada



Challenge:

- Statistics Canada **wants** to adopt open-source programming languages

Challenge:

- Statistics Canada **wants** to adopt open-source programming languages

...**but**...

- Legal obligation to ensure that personal information is kept **strictly** confidential

Challenge:

- Statistics Canada **wants** to adopt open-source programming languages

...**but**...

- Legal obligation to ensure that personal information is kept **strictly** confidential

...**therefore**...

- **G-Confid** (internal system) is used; prevents the release of confidential information

Challenge:

- Statistics Canada **wants** to adopt open-source programming languages

...**but**...

- Legal obligation to ensure that personal information is kept **strictly** confidential

...**therefore**...

- **G-Confid** (internal system) is used; prevents the release of confidential information

...**and**...

- G-Confid is **only** implemented in the SAS programming language

My typical workflow (now)



SAS dataset
+pre-processing

My typical workflow (now)



**SAS dataset
+pre-processing**

Analysis in R

My typical workflow (now)



**SAS dataset
+pre-processing**

Analysis in R

G-Config

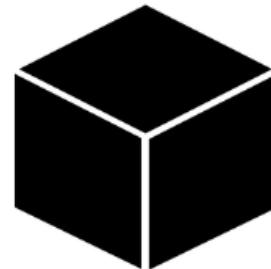
My typical workflow (now)



**SAS dataset
+pre-processing**



Analysis in R



G-Confid



Data viz in R

Solution:

Analysts at Statistics Canada need a way to combine R, Python, and SAS in one script, in a way that doesn't cause a headache!



The 3 step recipe

Step 1: Install the following



Step 2: Install SASPy

- **SASPy** is an (open source) Python package created by the SAS Foundation

```
reticulate::py_install("saspy", pip = TRUE)
```

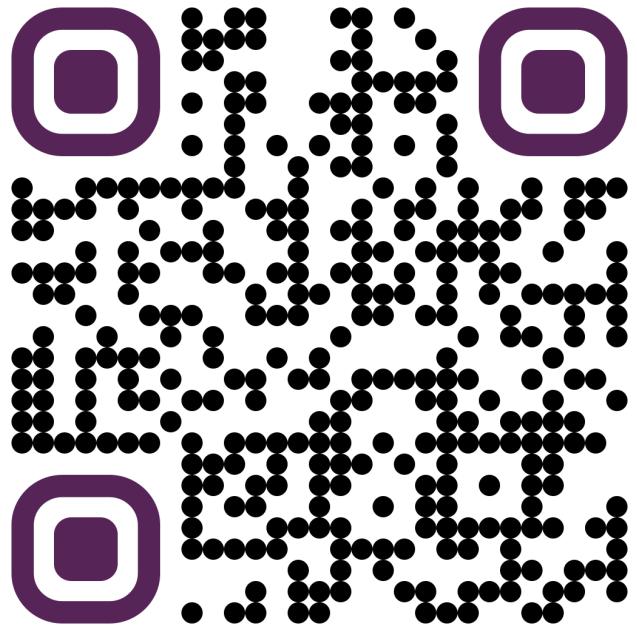


Step 3: System configurations

- Navigate to the **site-packages** folder where your Python installation is located
- Copy the **sascfg.py** file and rename the copy to **sascfg_personal.py**
- Switch which of the SAS_config_names lines is commented out:

```
# |SAS_config_names = ['default', 'ssh', 'iomlinux', 'iomwin',
# 'winlocal', 'winiomlinux', 'winiomwin', 'httpsviya', 'httpviya',
# 'iomcom']
#
SAS_config_names=['default']
```

- Add `java` as an Environment Variable on your computer



What to do when you don't have all the ingredients

Option 1: The SASmarkdown package

Provides interaction between SAS+R? 

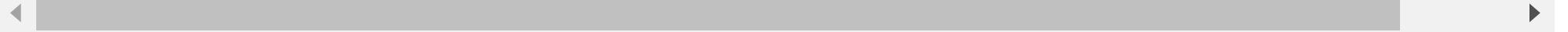
Can do without admin privileges? 

Usefulness rating:    

The SASmarkdown package allows you to include SAS chunks in your RMarkdown documents.

Option 1: The SASmarkdown package

```
```{r, engine="sashtml", engine.path="C:/Program Files/SASHome/SASFoundation/9.4/sa  
proc means data=sashelp.class;
run;
```
```



Option 1: The SASmarkdown package

```
```{r, engine="sashtml", engine.path="C:/Program Files/SASHome/SASFoundation/9.4/sa  
proc means data=sashelp.class;
run;
```
```



```
```{r}  
plot(cars)
```
```

Option 1: The SASmarkdown package

```
```{r, engine="sashtml", engine.path="C:/Program Files/SASHome/SASFoundation/9.4/sa  
proc means data=sashelp.class;
run;
```
```



```
```{r}  
plot(cars)
```
```

```
```{python}  
print("Hello world!"[0:5])
```
```

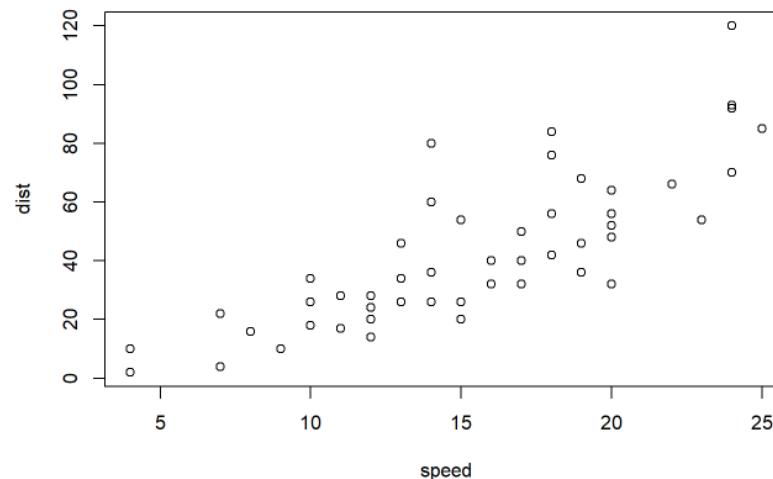
```
proc means data=sashelp.class;  
run;
```

SAS chunk

| Variable | N | Mean | Std Dev | Minimum | Maximum |
|----------|----|-------------|------------|------------|-------------|
| Age | 19 | 13.3157895 | 1.4926722 | 11.0000000 | 16.0000000 |
| Height | 19 | 62.3368421 | 5.1270752 | 51.3000000 | 72.0000000 |
| Weight | 19 | 100.0263158 | 22.7739335 | 50.5000000 | 150.0000000 |

```
plot(cars)
```

R chunk



```
print("Hello world"[0:5])
```

```
## Hello
```

Python chunk

Option 2: The RLANG option

Provides interaction between SAS+R? 

Can do without admin privileges? 

Usefulness rating: 

Option 2: The RLANG option

proc iml;

Provides interaction between SAS+R? 

Can do without admin privileges? 

Usefulness rating:  submit;

Option 2: The RLANG option

Provides interaction between SAS+R? 

Can do without admin privileges? 

Usefulness rating: 

```
proc iml;  
  submit / R;  
  
  ends submit;  
  submit;
```

Option 2: The RLANG option

Provides interaction between SAS+R? 

Can do without admin privileges? 

Usefulness rating: 

```
proc iml;  
  submit / R;  
    # Your R code goes here  
  endssubmit;  
  submit;
```

Option 2: The RLANG option

Provides interaction between SAS+R? 

Can do without admin privileges? 

Usefulness rating: 

```
proc iml;  
  submit / R;  
    # Your R code goes here  
  endssubmit;  
  submit;
```

To move data between R and SAS:

```
call ImportDataSetFromR("sas_data_name", "r_data_name");
```

Option 2: The RLANG option

Provides interaction between SAS+R? 

Can do without admin privileges? 

Usefulness rating: 

```
proc iml;  
  submit / R;  
    # Your R code goes here  
  endssubmit;  
  submit;
```

To move data between R and SAS:

```
call ImportDataSetFromR("sas_data_name", "r_data_name");
```

```
call ExportDataSetToR("sas_data_name", "r_data_name");
```

Option 2: The RLANG option

Provides interaction between SAS+R?

Can do without admin privileges?

Usefulness rating:

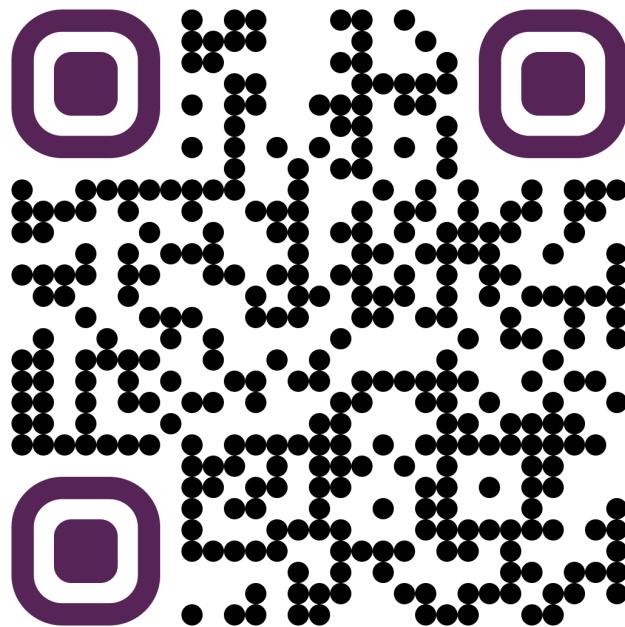
```
proc iml;  
  submit / R;  
    # Your R code goes here  
  endsubmit;  
  submit;
```

To move data between R and SAS:

```
call ImportDataSetFromR("sas_data_name", "r_data_name");
```

```
call ExportDataSetToR("sas_data_name", "r_data_name");
```

Option 2: The RLANG option



How to run R code from
within SAS Enterprise Guide

1.2K views • 1 year ago

Option 3: Command line

Provides interaction between SAS+R? 

Can do without admin privileges? 

Usefulness rating: 

Option 3: Command line

Provides interaction between SAS+R? 

Can do without admin privileges? 

Usefulness rating: 

```
SAS_code <- "proc freq data=work.in_df; tables cyl / out=work.out_df; run;"  
manipulate_df_in_SAS(in_df = mtcars,  
                     SAS_location = "C:/Program Files/SASHome/SASFoundation/9.4",  
                     SAS_code = SAS_code)
```

Option 3: Command line

Provides interaction between SAS+R? 

Can do without admin privileges? 

Usefulness rating: 

```
SAS_code <- "proc freq data=work.in_df; tables cyl / out=work.out_df; run;"  
manipulate_df_in_SAS(in_df = mtcars,  
                      SAS_location = "C:/Program Files/SASHome/SASFoundation/9.4",  
                      SAS_code = SAS_code)
```

Option 3: Command line

Provides interaction between SAS+R? 

Can do without admin privileges? 

Usefulness rating:     

```
SAS_code <- "proc freq data=work.in_df; tables cyl / out=work.out_df; run;"  
manipulate_df_in_SAS(in_df = mtcars,  
                     SAS_location = "C:/Program Files/SASHome/SASFoundation/9.4",  
                     SAS_code = SAS_code)
```

Option 3: Command line

Provides interaction between SAS+R? 

Can do without admin privileges? 

Usefulness rating: 

```
SAS_code <- "proc freq data=work.in_df; tables cyl / out=work.out_df; run;"  
manipulate_df_in_SAS(in_df = mtcars,  
                     SAS_location = "C:/Program Files/SASHome/SASFoundation/9.4",  
                     SAS_code = SAS_code)
```

Option 3: Command line

Provides interaction between SAS+R? 

Can do without admin privileges? 

Usefulness rating: 

```
SAS_code <- "proc freq data=work.in_df; tables cyl / out=work.out_df; run;"  
manipulate_df_in_SAS(in_df = mtcars,  
                     SAS_location = "C:/Program Files/SASHome/SASFoundation/9.4",  
                     SAS_code = SAS_code)
```

Option 3: Command line

Provides interaction between SAS+R? 

Can do without admin privileges? 

Usefulness rating: 

```
SAS_code <- "proc freq data=work.in_df; tables cyl / out=work.out_df; run;"  
manipulate_df_in_SAS(in_df = mtcars,  
                     SAS_location = "C:/Program Files/SASHome/SASFoundation/9.4",  
                     SAS_code = SAS_code)
```

Option 3: Command line

Provides interaction between SAS+R? 

Can do without admin privileges? 

Usefulness rating: 

```
SAS_code <- "proc freq data=work.in_df; tables cyl / out=work.out_df; run;"  
manipulate_df_in_SAS(in_df = mtcars,  
                     SAS_location = "C:/Program Files/SASHome/SASFoundation/9.4",  
                     SAS_code = SAS_code)
```

| cyl | COUNT | PERCENT |
|-----|-------|---------|
| 4 | 11 | 34.375 |
| 6 | 7 | 21.875 |
| 8 | 14 | 43.750 |

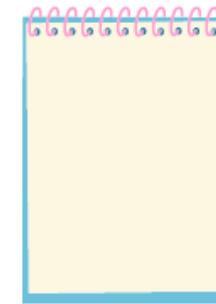
Option 3: Command line



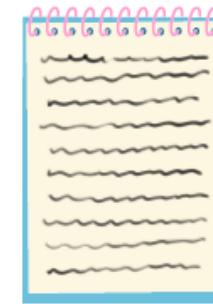
Option 3: Command line



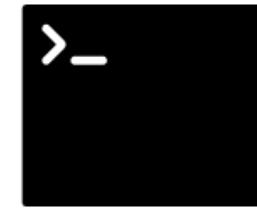
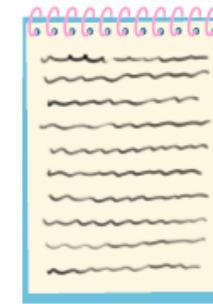
Option 3: Command line



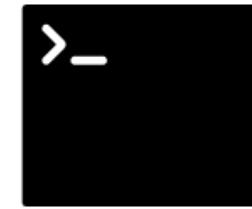
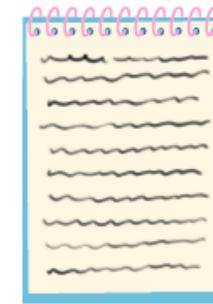
Option 3: Command line



Option 3: Command line



Option 3: Command line



Option 3: Command line



Conclusions...









1. SASmarkdown package



- 1. SASmarkdown package**
- 2. RLANG option**



- 1. SASmarkdown package**
- 2. RLANG option**
- 3. DIY command line**



- 1. SASmarkdown package**
- 2. RLANG option**
- 3. DIY command line**



- 1. SASmarkdown package**
- 2. RLANG option**
- 3. DIY command line**



melissavanbussel.com



linkedin.com/in/melissavanbussel



@ggnot2



@melissavanbussel