

# Getting data into Rstudio and SAS/EG

## Descriptive statistics

ABB lecture series – lecture 2

# Preparing data files

- <https://intranet.psb.ugent.be/applications> -> Statistical Consulting Service

## Data format and sharing

Before importing data in a statistical software package, the data must be well prepared to avoid errors. **Column names** should be compatible with naming conventions. Please read the guidelines. Each row should contain all data measured on one plant.

The data must be saved as a **.csv file** (comma separated file). Please give it the following name "**yyyymmdd\_username\_keyword.csv**" (eg 20180629\_asgad\_AZD.csv).

A second csv file should be provided containing the **metadata** and named "**yyyymmdd\_username\_keyword\_meta.csv**". This file should have in the first column the variable names, in a second column the meaning of the numbers. If a character variable is numerically coded, the meaning of the numbers should be given as well.

More information on data formatting can be found on the biostat wiki page.

Prior to uploading your data, **run the application** on <https://intranet.psb.ugent.be/shiny/importCSV/>

**Verify** that

- column headers have the proper name.
- categorical variables have the expected number of levels. Non-consistent names
- Look at min and max values of numerical variables. Are they within the expected range?



**Best Practices in Preparing Data Files for Importing into R**

If all is well, datasets can be copied to <..\psb-department\stathelp>.

# Shiny application

https://intranet.psb.ugent.be/shiny/importCSV/

Apps ABB arab atletiek BIO-IT conferences google HOME jmp JOBS lit maize MOBILITI

## Uploading Files

Choose CSV File

Browse... 20181001\_alalv\_Benzodioxane.csv

Upload complete

Header

**Separator**

Comma

Semicolon

Tab

**Quote**

None

Double Quote

Single Quote

**Display**

Head

All

Browse for file

GENOTYPE	Height_15weeks_cm
comt1 comt2	167.00
comt1 comt2	178.00
comt1 comt2	208.00
comt1 comt2	196.00
comt1 comt2	166.00
comt1 comt2	160.00

# Shiny application

		Table	Structure	Summary			
No	Variable	Stats / Values	Freqs (% of Valid)	Text Graph	Valid	Missing	
1	GENOTYPE [factor]	1. C4H:F5H1	14 (19.7%)	IIIIIIIIIIII	71	0 (100%) (0%)	
		2. comt1	6 ( 8.5%)	IIII			
		3. comt1 comt2	16 (22.5%)	IIIIIIIIIIIIII			
		4. comt1 comt2 C4H:F5H1	18 (25.4%)	IIIIIIIIIIIIIIII			
		5. comt2	2 ( 2.8%)	I			
		6. WT	15 (21.1%)	IIIIIIIIIIII			
2	Height_15weeks_cm [numeric]	mean (sd) : 175.73 (44.83)	58 distinct val.	. . .	71	0 (100%) (0%)	
		min < med < max :		: : : :			
		40 < 188 < 232		: : : :			
		IQR (CV) : 49 (0.26)		. . : : : :			
				: : . : : : :			

# SAS versus R

## SAS

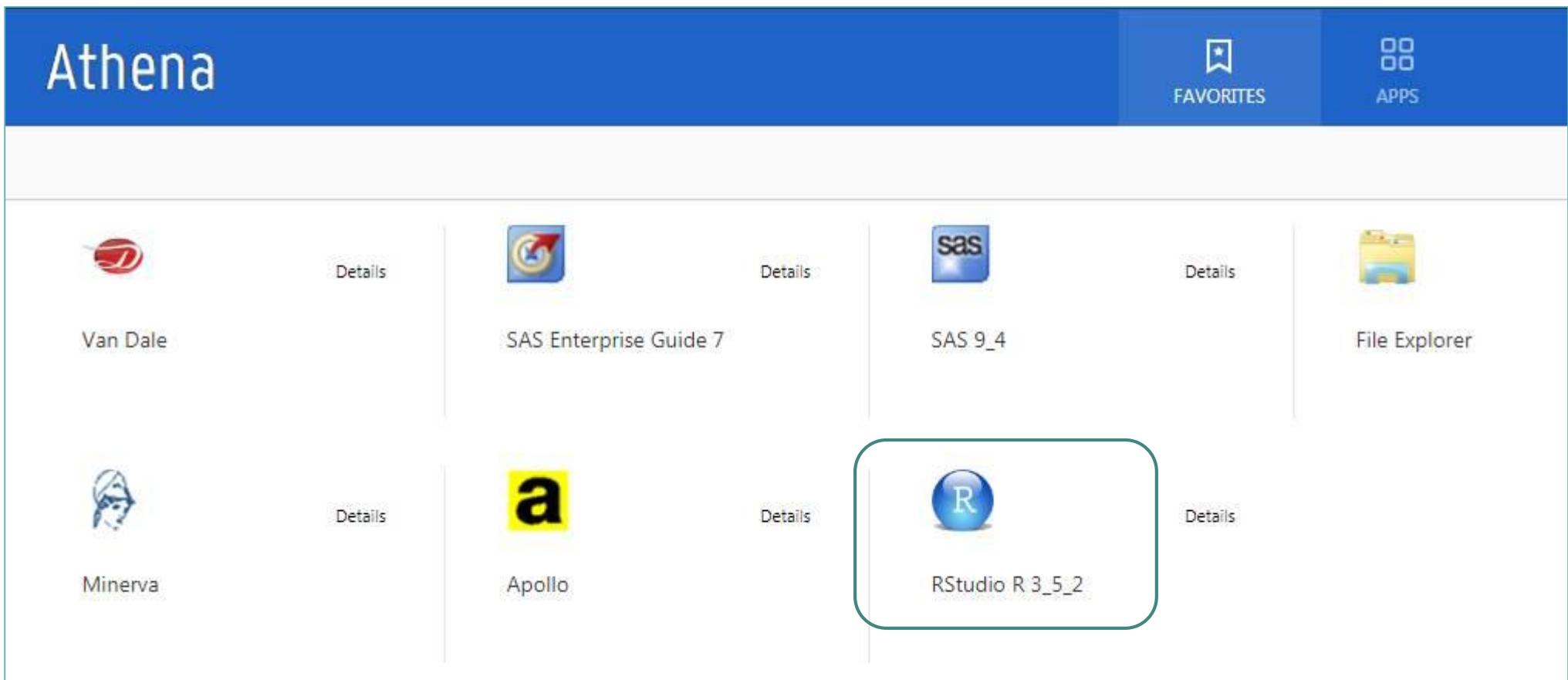
Strengths	weaknesses
Large and complex datasets	Expensive, but now free SAS university edition for academic and non-commercial use <a href="https://www.sas.com/en_us/software/university-edition.html">https://www.sas.com/en_us/software/university-edition.html</a>
Powerful graphics tools	Graphics can be difficult to learn
Mixed models	
Good documentation	
FDA approved	

## R

Strengths	weaknesses
free	Steep learning curve
Easy to use graphics	Limited documentation
Analyses packages are constantly created	Strength of programs depends on the developer

# R and RStudio

# Getting RStudio



R Studio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Console Terminal

H:/

```
R version 3.5.2 (2018-12-20) -- "Eggshell Igloo"
Copyright (C) 2018 The R Foundation for Statistical computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
```

> |

Environment History Connections

Import Dataset Global Environment

Environment is empty

Files Plots Packages Help Viewer

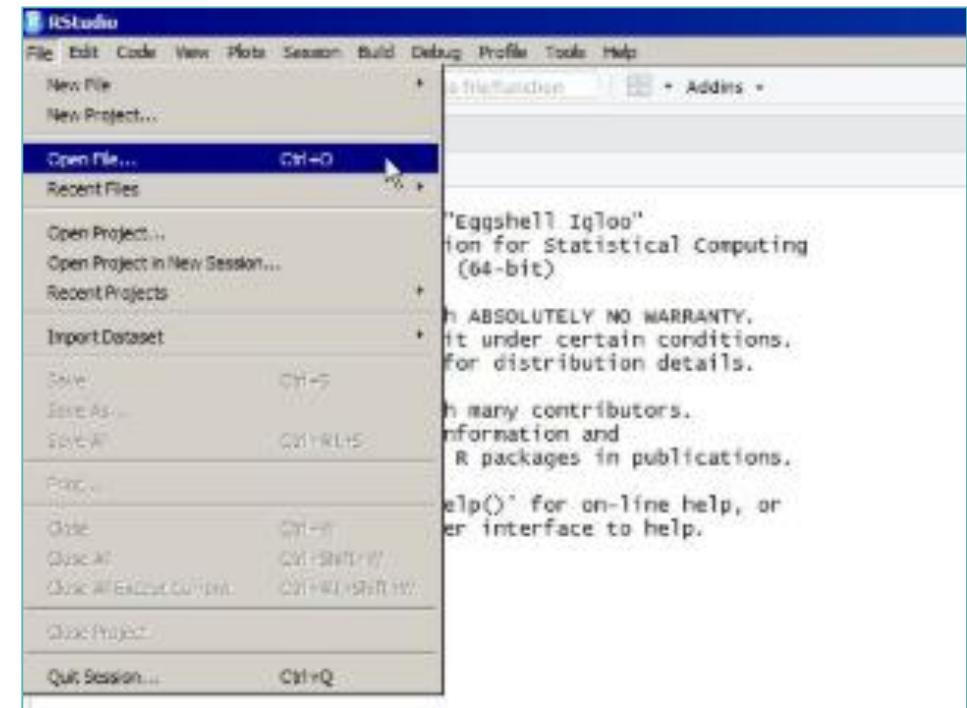
New Folder Delete Rename More

H:

	Name	Size	Modified
	..		
	.Rhistory	17 KB	Jun 4, 2018, 3:21 PM
	Athena		
	Custom Office Templates		
	Desktop		
	Diamonds		
	Downloads		
	Favorites		
	home (filesvstorne) (H) - Shortcut.lnk	590 B	Aug 29, 2018, 9:28 PM
	My Music		
	My Pictures		
	My SAS Files		
	My Videos		
	My_SAS_Files		
	price_evolution.png	25.4 KB	Jan 2, 2019, 3:13 PM
	R		
	SafeNet Sentinel		
	SAS94		
	SAScolornames.txt	103 B	Aug 30, 2018, 9:42 PM
	SASprogrI		
	SavedPlot.pdf	4.3 KB	Nov 13, 2018, 4:22 PM
	SigmaPlot		

# Open an existing R script

- File
  - ▶ Open File



# Set a working directory

- Create a folder (in the usual way in Windows or MacOS) and place the data files you'll be using in that folder.
- When you start your session in R change the working directory to be the folder that you have just created.

# Set a working directory

- Let's assume that you have a folder in 'ANALYSES' called 'mySMALLprojects' with subfolder '2018' and within that you have created a folder called 'alalv' in which you have placed some data files that you want to analyse.
- Set the working directory using ***setwd()*** :
  - ▶ On a local drive
    - `setwd("//Client/C$/ANALYSES/mySMALLprojects/2018/alalv")`
  - ▶ On a network drive
    - `setwd("L:/bioit/groups/bioit.biostat/ANALYSES/mySMALLprojects/2018/alalv")`
  - A listing of the files in that directory can be gotten with ***dir()***

# Installing and loading R packages

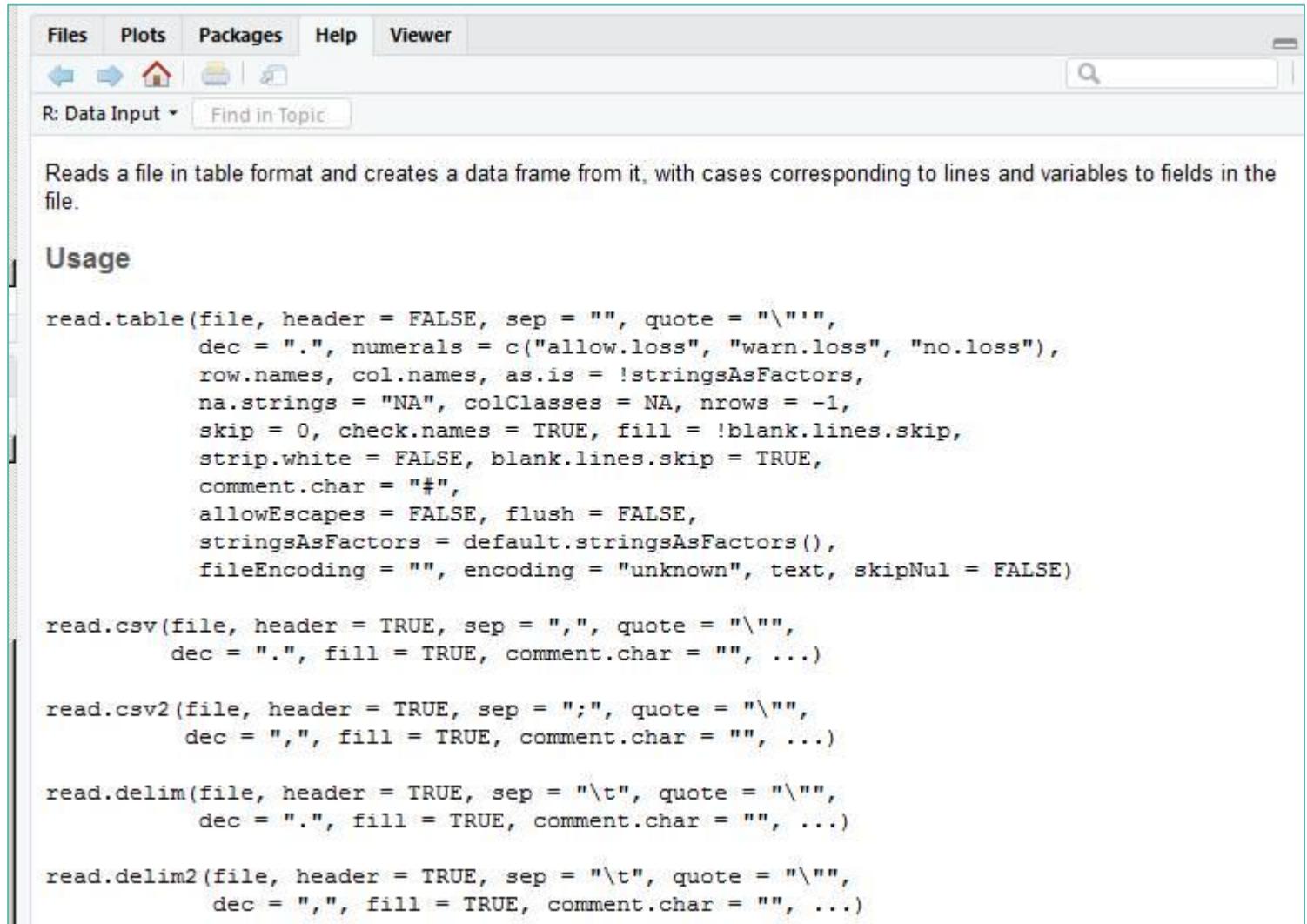
- R packages have been pre-installed
  - ▶ Packages cannot be installed when you are working from Athena
- Loading R packages in memory
  - ▶ `library(plyr)`

# Importing data

- Comma separated data with “.” as decimal separator
  - ▶ `mydata <- read.csv(file = "filename.csv")`
- Semi-colon separated data with “;” as decimal separator
  - ▶ `mydata <- read.csv2(file = "filename.csv")`
- Tab delimited data with “.” as decimal separator
  - ▶ `mydata <- read.delim(file = "filename.txt")`
- Tab delimited data with “;” as decimal separator
  - ▶ `mydata <- read.delim2(file = "filename.txt")`
- Missing data is assumed blank or NA

# Getting help on a function

► ?read.csv



The screenshot shows the RStudio interface with the 'Help' tab selected in the top menu bar. Below the menu is a toolbar with icons for file operations like Open, Save, and Home. A search bar is located on the right side of the toolbar. The main area displays the help documentation for the 'read.csv' function. The title of the help page is 'R: Data Input' and there is a 'Find in Topic' button. The first section, 'Usage', contains the R code for 'read.table', 'read.csv', 'read.csv2', 'read.delim', and 'read.delim2'. The code uses backticks to indicate the function names.

```
read.table(file, header = FALSE, sep = "", quote = "\t",
          dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),
          row.names, col.names, as.is = !stringsAsFactors,
          na.strings = "NA", colClasses = NA, nrows = -1,
          skip = 0, check.names = TRUE, fill = !blank.lines.skip,
          strip.white = FALSE, blank.lines.skip = TRUE,
          comment.char = "#",
          allowEscapes = FALSE, flush = FALSE,
          stringsAsFactors = default.stringsAsFactors(),
          fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)

read.csv(file, header = TRUE, sep = ",", quote = "\t",
         dec = ".", fill = TRUE, comment.char = "", ...)

read.csv2(file, header = TRUE, sep = ";", quote = "\t",
          dec = ",", fill = TRUE, comment.char = "", ...)

read.delim(file, header = TRUE, sep = "\t", quote = "\t",
           dec = ".", fill = TRUE, comment.char = "", ...)

read.delim2(file, header = TRUE, sep = "\t", quote = "\t",
            dec = ",", fill = TRUE, comment.char = "", ...)
```

# Reading data from excel

```
# Loading
  ▶ library("readxl")

# xls files
  ▶ mydata <- read_excel("my_file.xls")

# xlsx files
  ▶ mydata <- read_excel("my_file.xlsx")

# Specify sheet by its name
  ▶ mydata <- read_excel("my_file.xlsx", sheet = "data")

# Specify sheet by its index
  ▶ mydata <- read_excel("my_file.xlsx", sheet = 2)

# If NAs are represented by something other than blank cells:
  ▶ mydata <- read_excel("my_file.xlsx", na = "99")
```

# Viewing data

- In a separate view window
  - ▶ `View(mydata)`
- In the console
  - ▶ `mydata`
- List the first 6 lines in the console
  - ▶ `head(mydata)`
- Access a variable from a data.frame
  - ▶ `mydata$varname`

# Data exploration

- Get the number of levels of a factor variable
  - ▶ `nlevels(mydata$factorvar)`
- Get the levels of a factor variable
  - ▶ `levels(mydata$factorvar)`
- Get the frequencies of each level
  - ▶ `table(mydata$factorvar)`

# Frequency tables

- Use frequency tables to
  - ▶ screen for invalid, missing, and duplicate data values.
  - ▶ verify the balancedness in the data

# Frequency tables

- One-way table
  - ▶ **table(mydata\$factorvar1)**
  - ▶ **gmodels::CrossTable(mydata\$factorvar1)**
- Two-way table
  - ▶ **table(mydata\$factorvar1, mydata\$factorvar2)**
  - ▶ **gmodels::CrossTable(mydata\$factorvar1, mydata\$factorvar2)**
- Multi -way table
  - ▶ **xtabs(~ factorvar1 + factorvar2 + factorvar3, data=mydata)**
  - ▶ **janitor::tabyl(mydata, factorvar1 , factorvar2 , factorvar3)**

```
table(mydata$treat)

##
## control      npa
##    280       305

table(mydata$gt, mydata$treat)

##
##                      control npa
## erf115oe            53   64
## luh-3               60   59
## luh-4               57   62
## seu-4              54   57
## wt                  56   63
```

```
CrossTable(mydata$gt, mydata$treat)
```

```
##  
##  
## Cell Contents  
## |-----|  
## | N |  
## Chi-square contribution  
## | N / Row Total |  
## | N / Col Total |  
## | N / Table Total |  
## |-----|  
##  
##  
## Total Observations in Table: 585  
##
```

## ## mydata\$gt	mydata\$treat		
	control	npa	Row Total
## erf115oe	53	64	117
##	0.161	0.148	
##	0.453	0.547	0.200
##	0.189	0.210	
##	0.091	0.109	
## luh-3	60	59	119
##	0.163	0.149	
##	0.504	0.496	0.203
##	0.214	0.193	
##	0.103	0.101	
## luh-4	57	62	119
##	0.000	0.000	
##	0.479	0.521	0.203
##	0.204	0.203	
##	0.097	0.106	
## seu-4	54	57	111
##	0.014	0.013	
##	0.486	0.514	0.190
##	0.193	0.187	
##	0.092	0.097	
## wt	56	63	119
##	0.016	0.015	
##	0.471	0.529	0.203
##	0.200	0.207	
##	0.096	0.108	
## Column Total	280	305	585
##	0.479	0.521	

```

xtabs(~ gt + treat +expID, data=mydata)

## , , expID = 1
##
##          treat
## gt      control npa
## erf115oe    11 22
## luh-3       16 21
## luh-4       16 19
## seu-4       17 21
## wt          19 20
##
## , , expID = 2
##
##          treat
## gt      control npa
## erf115oe    19 21
## luh-3       19 18
## luh-4       20 23
## seu-4       18 17
## wt          23 22
##
## , , expID = 3
##
##          treat
## gt      control npa
## erf115oe    23 21
## luh-3       25 20
## luh-4       21 20
## seu-4       19 19
## wt          14 21

```

```

tabyl(mydata, gt, treat, expID)

## $`1`
##      gt control npa
## erf115oe    11 22
## luh-3       16 21
## luh-4       16 19
## seu-4       17 21
## wt          19 20
##
## $`2`
##      gt control npa
## erf115oe    19 21
## luh-3       19 18
## luh-4       20 23
## seu-4       18 17
## wt          23 22
##
## $`3`
##      gt control npa
## erf115oe    23 21
## luh-3       25 20
## luh-4       21 20
## seu-4       19 19
## wt          14 21

```

# Descriptive statistics

- Many available functions
  - ▶ `summary()` from base R
  - ▶ `ddply()` from the `plyr` package
  - ▶ `summary_stats()` from the `summarizer` package
  - ▶ `ci.mean()` from the `Publish` package
  - ▶ `favstats()` from the `mosaic` package
  - ▶ `describeBy()` from the `psych` package
  - ▶ `stat.desc()` from the `pastecs` package
  - ▶ `tableby()` from the `arsenal` package
  - ▶ ...

# Base

▶ summary(mydata)

```
##      expID          gt       treat      seedling
##  Min.   :1.00  erf115oe:117  control:280  Min.   : 1.0
##  1st Qu.:1.00    luh-3   :119    npa     :305  1st Qu.: 5.0
##  Median :2.00    luh-4   :119           Median :10.0
##  Mean   :2.04    seu-4   :111           Mean   :10.5
##  3rd Qu.:3.00    wt      :119           3rd Qu.:15.0
##  Max.   :3.00
##  amyloplastheight
##  Min.   : 44
##  1st Qu.: 62
##  Median : 91
##  Mean   :140
##  3rd Qu.:210
##  Max.   :475
```

# plyr package

- ▶ 

```
ddply.twoway <- ddply(mydata, .(gt, treat), summarize,
  N = length(amyloplastheight),
  Nmiss = sum(is.na(amyloplastheight)),
  mean = mean(amyloplastheight),
  sd = sd(amyloplastheight),
  se  = sd/sqrt(N),
  t = qt(0.975, N-1),
  lower = mean - t*se,
  upper = mean + t*se)
```
- ▶ # print the df with 2 decimals for the mean and the CI
- ▶ 

```
ddply.twoway[,5:10] <- round(ddply.twoway[,5:10],2)
```
- ▶ 

```
ddply.twoway
```

# plyr package

##	gt	treat	N	Nmiss	mean	sd	se	t	lower	upper
## 1	erf115oe	control	53	0	67	9.4	1.29	2	64	69
## 2	erf115oe	npa	64	0	266	85.3	10.67	2	245	287
## 3	luh-3	control	60	0	60	7.0	0.90	2	58	61
## 4	luh-3	npa	59	0	218	102.6	13.35	2	191	245
## 5	luh-4	control	57	0	60	6.2	0.83	2	58	62
## 6	luh-4	npa	62	0	188	103.6	13.15	2	162	214
## 7	seu-4	control	54	0	68	8.0	1.09	2	66	70
## 8	seu-4	npa	57	0	241	68.9	9.13	2	223	260
## 9	wt	control	56	0	58	6.7	0.90	2	56	60
## 10	wt	npa	63	0	148	53.5	6.74	2	135	162

# summariser package

- ▶ **summary\_stats**(data=mydata, measure = "amyloplastheight", gt)
- ▶ **summary\_stats**(data=mydata, measure = "amyloplastheight", gt, treat)

# Publish package

- ▶ `ci.mean(amyloplastheight ~ gt, data=mydata)`
- ▶ `ci.mean(amyloplastheight ~ gt + treat, data=mydata)`

# mosaic package

- ▶ **favstats(~ amyloplastheight | gt, data=mydata)**
- ▶ **favstats(~ amyloplastheight | gt + treat, data=mydata)**

# psych package

- ▶ **describeBy**(mydata\$amyloplastheight, mydata\$gt, mat=TRUE, digits=2)
- ▶ **describeBy**(mydata\$amyloplastheight, list(mydata\$gt, mydata\$treat), mat=TRUE, digits=2)

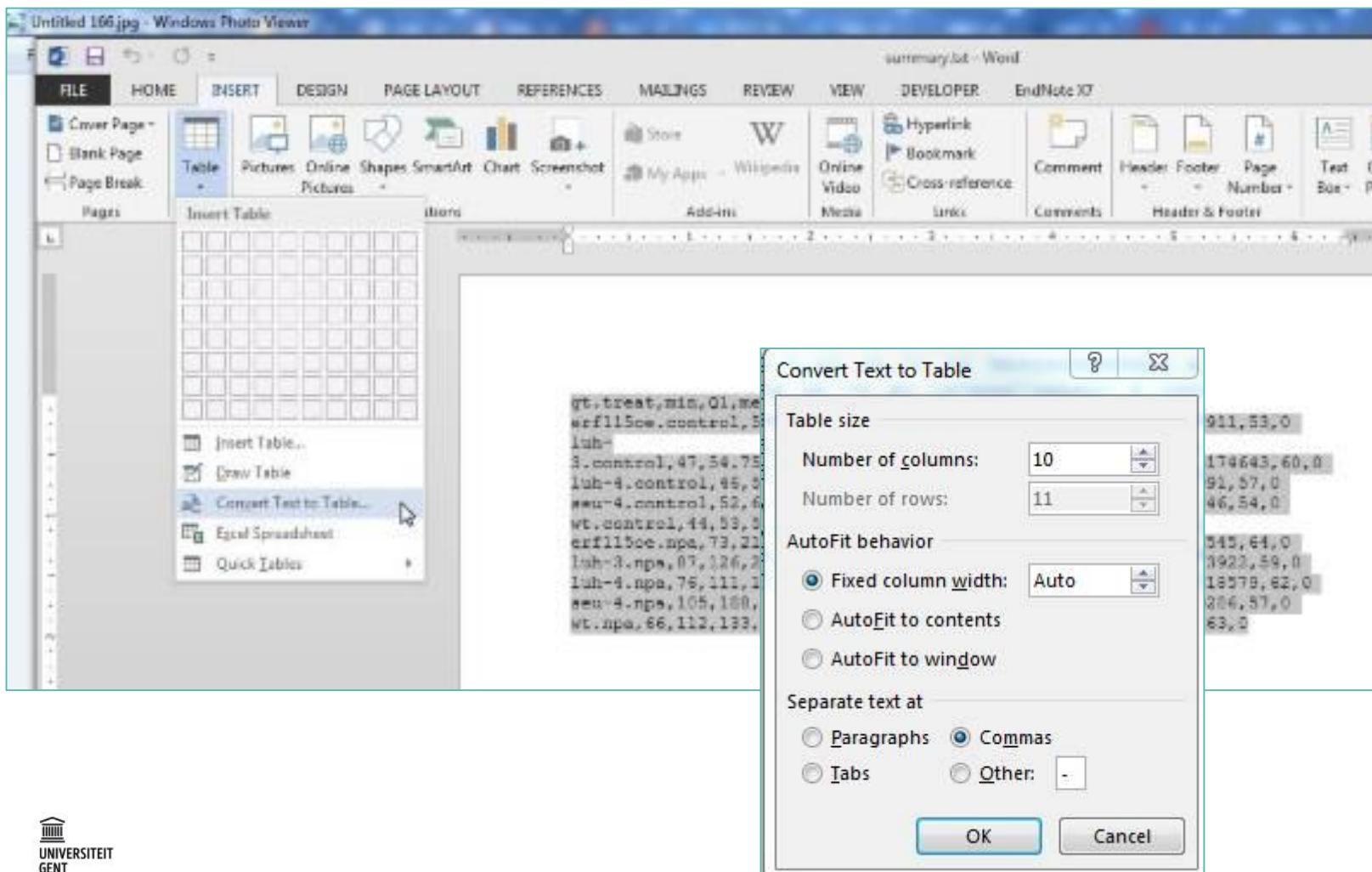
# pastecs package

- ▶ `desc.gt <- by(mydata$amyloplastheight, mydata$gt, stat.desc)`
- ▶ `do.call(rbind, desc.gt)`
  
- ▶ `desc.twoway <- by(mydata$amyloplastheight, list(mydata$gt, mydata$treat), stat.desc)`
- ▶ `do.call(rbind, desc.twoway)`

# Output the results

► **write.table(results, file = "filename.txt", sep = ",", quote = FALSE, row.names = F)**

# Create a table in word

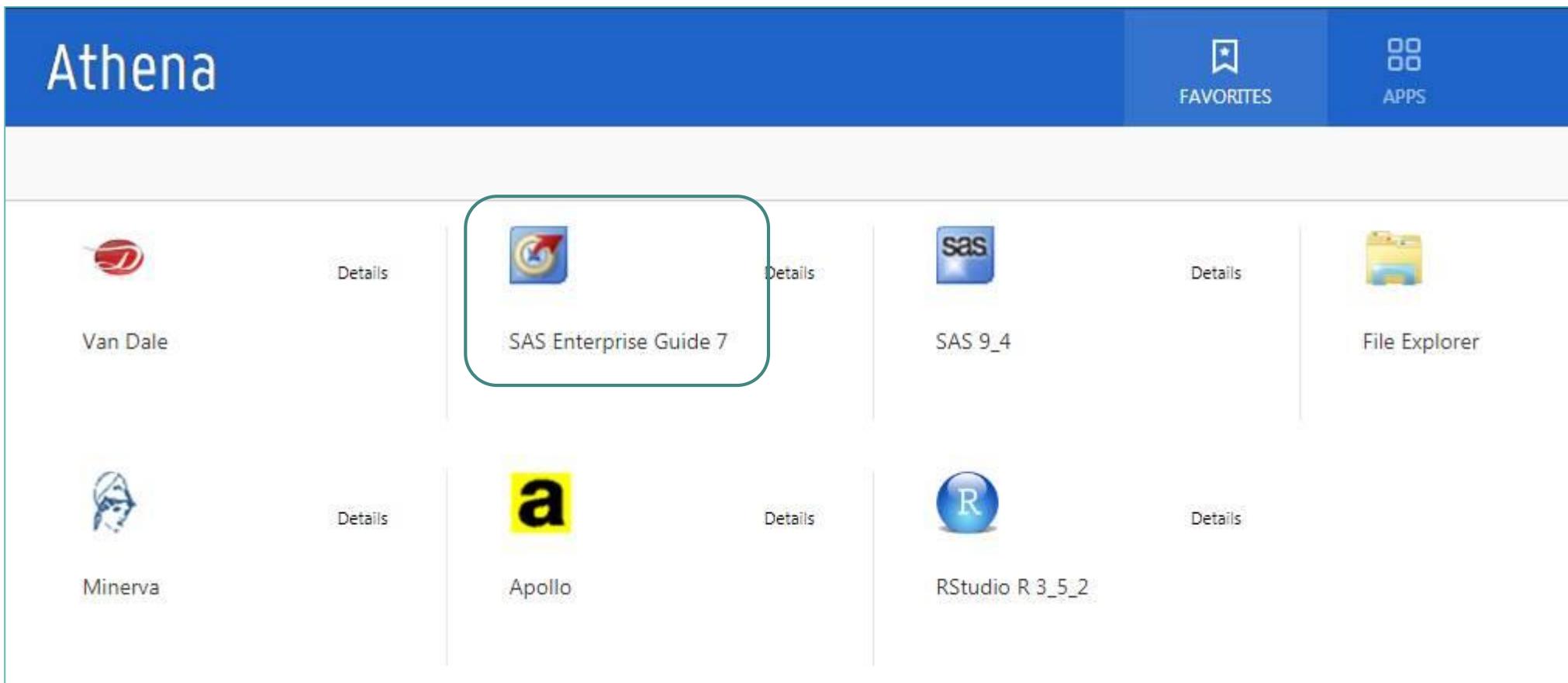


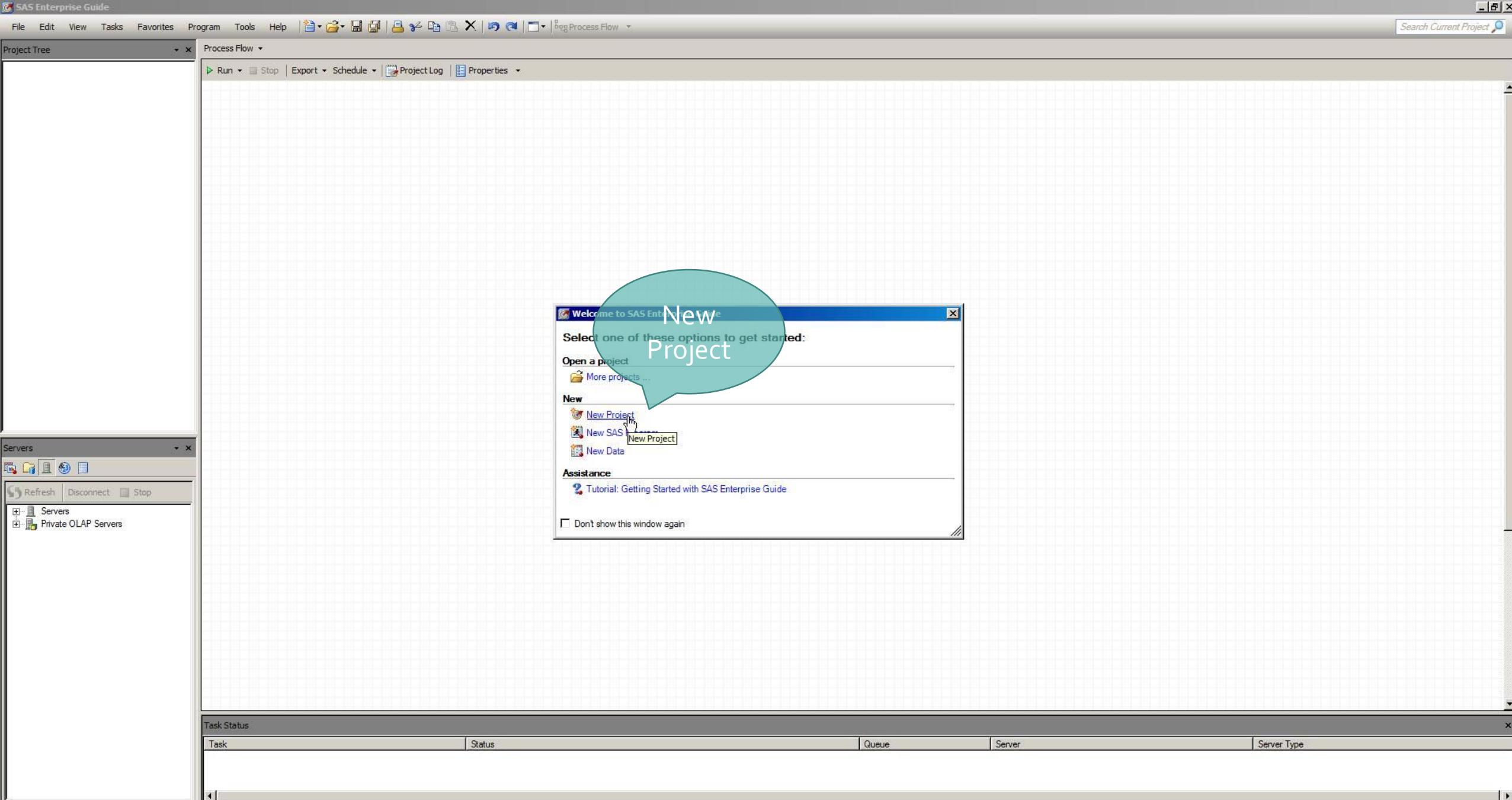
# Try it yourself

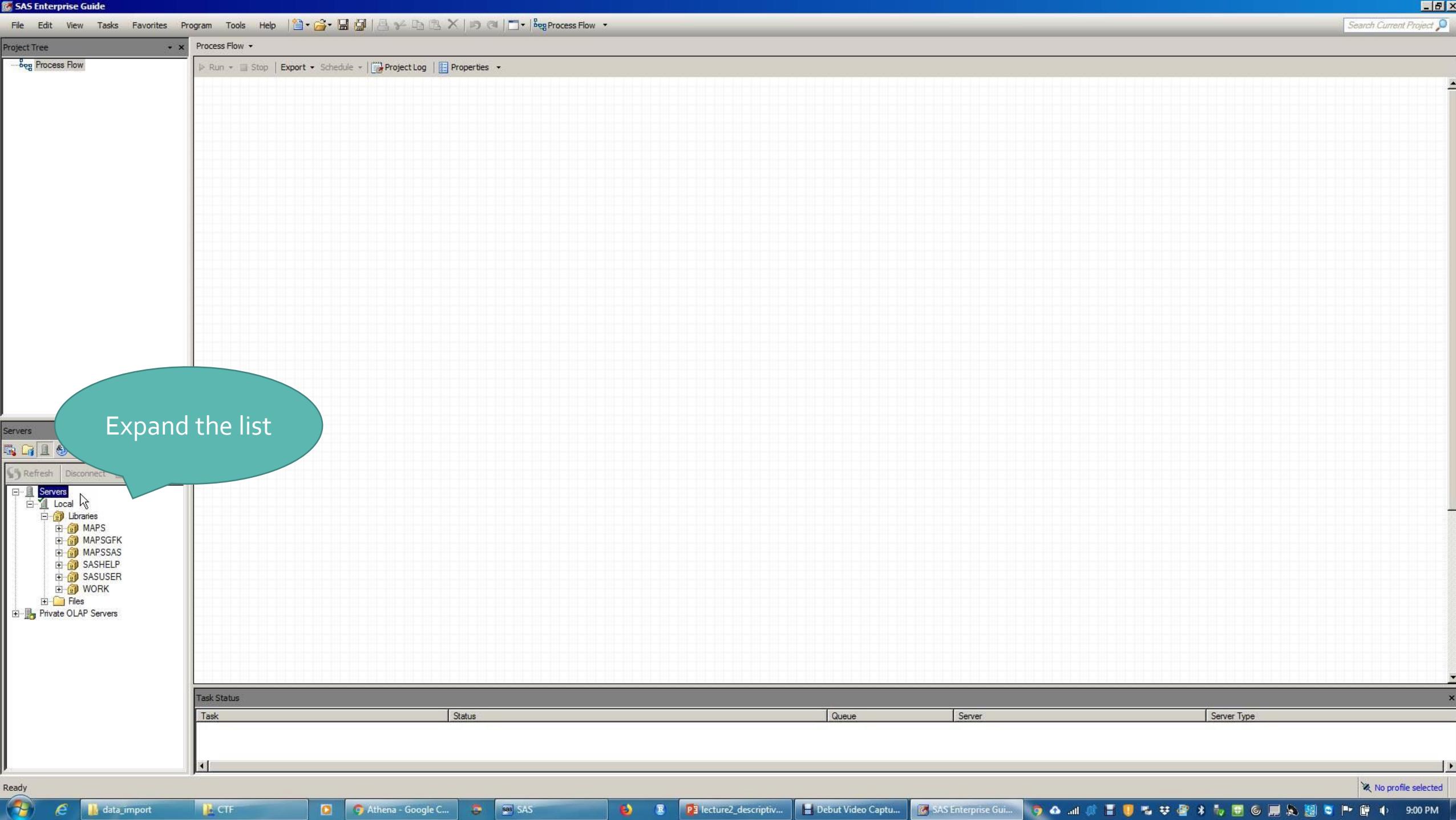
- Prepare some data
- Open Rstudio and run the R code **data\_import.R** available on  
\\psb.ugent.be\shares\psb-department\psb\_abb\STATISTICS\SCRIPTS\R

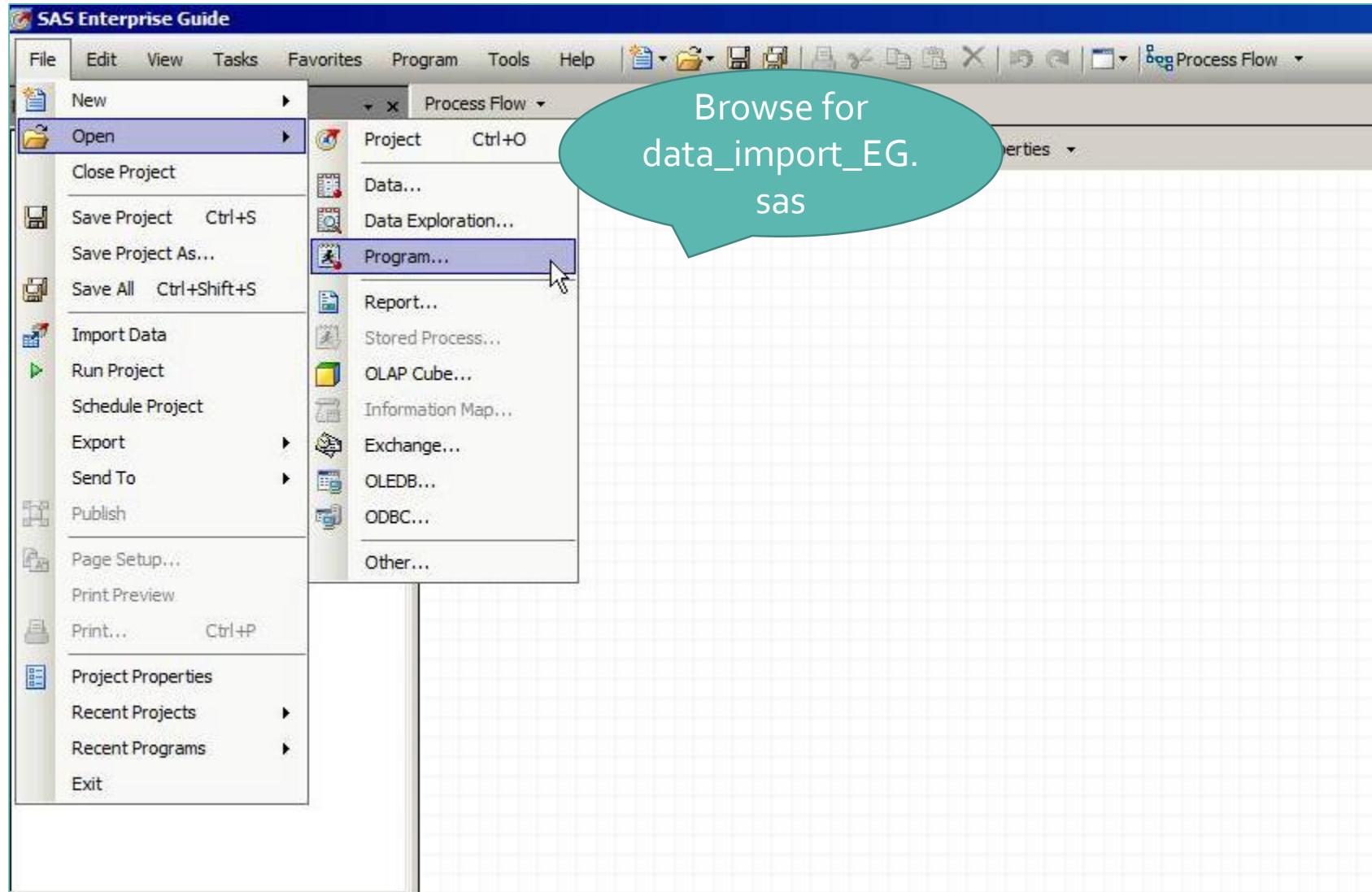
# SAS Enterprise Guide

# launch SAS Enterprise Guide









SAS Enterprise Guide

File Edit View Tasks Favorites Program Tools Help | Process Flow | Process Flow

Project Tree | data\_import

Process Flow | Programs | data\_import

Save | Run | Stop | Selected Server: Local | Analyze Program | Export | Send To | Create | Changes | Commit | History | Properties

\* set library  
\*\*\* Run Selection \*\*\*\*;  
libname demo '\\Client\C\$\MYCOURSES\VI Bstatseminars\data\_import';  
  
\* import the data  
\*\*\*\*\*;  
  
proc import datafile = "\\client\C\$\MYCOURSES\VI Bstatseminars\data\_import\20180420\_bacan\_amyloplast1.txt"  
out = demo.mydata  
dbms = tab replace ;  
datarow=2;  
getnames=yes;  
guessingrows=max;  
run;  
proc print data = mydata (obs=5) ;  
run;

# Setting the 'working directory'

```
libname demo '\\Client\C$\MYCOURSES\VIBstatseminars\data_import'
```

**LIBNAME** *libref*"*SAS-library*" <*options*>;

# Import excel file

- From excel2007 or later:

```
proc import out= demo.sasdsn  
    datafile= "path\filename.xlsx"  
    dbms=xlsx replace ;  
  
    getnames=yes;  
    datarow=2;  
    *sheet= "sheetname" ;  
  
run;
```

# Import tab delimited file

```
proc import out= work.sasdsn  
            datafile= "path\dsn.txt"  
            dbms=tab replace ;  
  
getnames=yes;  
datarow=2;  
guessingrows=max;  
  
run;
```

# Import csv file

```
proc import out= work.sasdsn  
            datafile= "path\dsn.csv"  
            dbms=csv replace ;  
  
getnames=yes;  
datarow=2;  
guessingrows=max;  
  
run;
```

# Import delimited file

- Files with delimiters other than commas or tabs

```
proc import out = WORK.sasdsn  
    datafile= "path\dsn.dat"  
    dbms=dlm replace ;  
    delimiter="/" ;  
    getnames=yes;  
    datarow=2;  
    guessingrows=max;  
  
run;
```

SAS Enterprise Guide

File Edit View Tasks Favorites Program Tools Help | | **Log Process Flow**

**Project Tree** **data\_import**

**Program Log**

Export Send To Create Project Log Properties

21:13 Tuesday, March 5, 2019

```

1      ;*';*/;quit;run;
2      OPTIONS PAGENO=MIN;
3      %LET _CLIENTTASKLABEL='data_import';
4      %LET _CLIENTPROCESSFLOWNAME='Process Flow';
5      %LET _CLIENTPROJECTPATH='';
6      %LET _CLIENTPROJECTPATHHOST='';
7      %LET _CLIENTPROJECTNAME='';
8      %LET _SASPROGRAMFILE='\\Client\C$\MYCOURSES\VIBstatseminars\data_import\data_import.sas';
9      %LET _SASPROGRAMFILEHOST='ATHA1212';
10
11     ODS _ALL_ CLOSE;
12     OPTIONS DEV=PNG;
13     GOPTIONS XPIXELS=0 YPIXELS=0;
14     FILENAME EGSR TEMP;
15     ODS tagsets.sasreport13(ID=EGSR) FILE=EGSR
16       STYLE=HtmlBlue
17       STYLESHEET=(URL="file:///C:/Program%20Files%20(x86)/SASClients/x86/SASEnterpriseGuide/7.1/Styles/HtmlBlue.css")
18       NOGTITLE
19       NOGFOOTNOTE
20       GPATH=&sasworklocation
21       ENCODING=UTF8
22       options(rolap="on")
23 ;
NOTE: Writing TAGSETS.SASREPORT13(EGSR) Body file: EGSR
24
25     GOPTIONS ACCESSIBLE;
26     libname demo '\\Client\C$\MYCOURSES\VIBstatseminars\data_import' ;

```

**Servers**

Refresh Disconnect Stop

- Servers
  - Local
    - DEMOS
    - MAPS
    - MAPSGFK
    - MAPSSAS
    - SASHelp
    - SASUSER
    - WORK
  - Files
  - Private OLAP Servers

**Log Summary**

Errors (0) Warnings (0) Notes (2)

Description	Line	Affected Code	Log Line	Program Line
NOTE: Writing TAGSETS.SASREPORT13(EGSR) Body file: EGSR	26	options(rolap="on")	22	
NOTE: Libref DEMO was successfully assigned as follows:	30	libname demo '\\Client\C\$\MYCOURSES\VIBstatseminars\data_import' ;	26	4

SAS Enterprise Guide

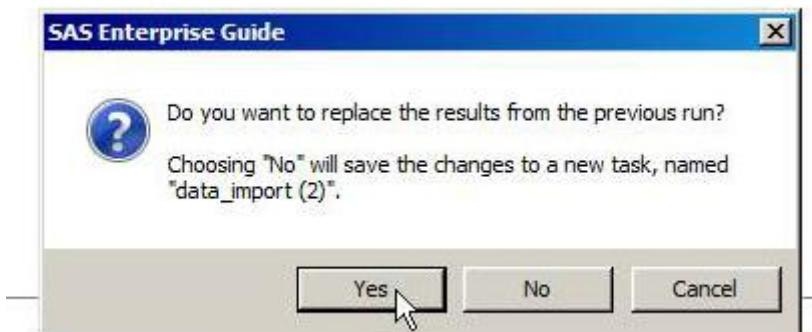
File Edit View Tasks Favorites Program Tools Help | | **Process Flow**

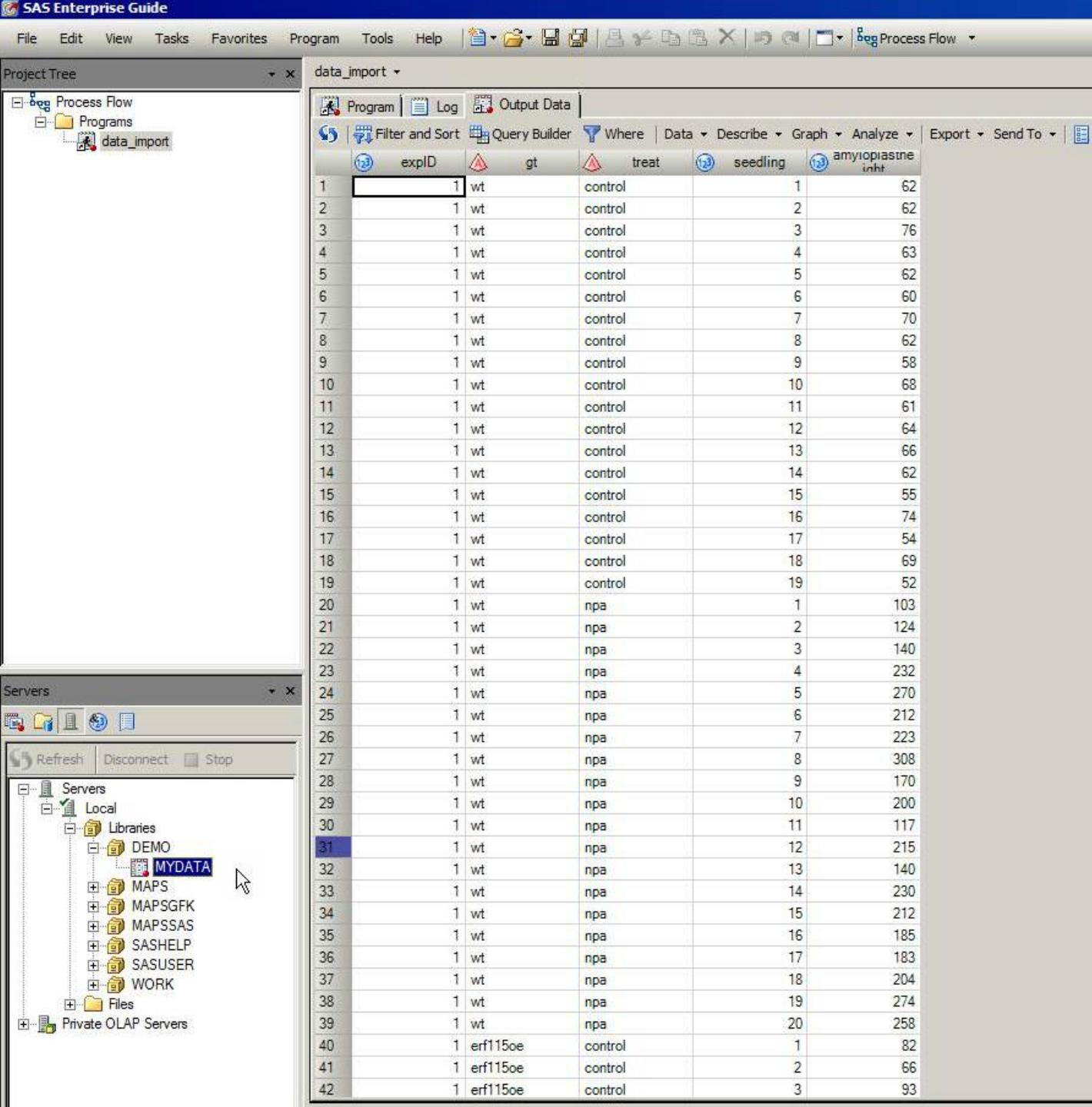
Project Tree **data\_import**

Program Log

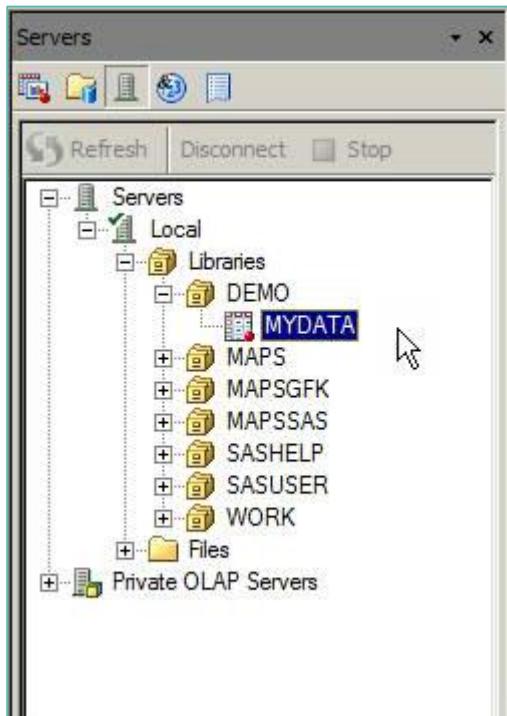
Save Run Stop Selected Server: Local (Connected) Analyze Program Export Send To Create Changes Commit History Properties

\* set library  
\*\*\* Run Selection \*\*\*\*;  
  
libname demo '\\Client\C\$\MYCOURSES\VIBstatseminars\data\_import' ;  
  
\* import the data  
\*\*\*\*\*;  
  
proc import datafile = "\\Client\C\$\MYCOURSES\VIBstatseminars\data\_import\20180420\_bacan\_amyloplast1.txt"  
out = demo.mydata  
dbms = tab replace ;  
datarow=2;  
getnames=yes;  
guessingrows=max;  
run;  
proc print data = mydata (obs=5) ;  
run;





# Viewing the file



```
proc print data = demo.mydata (obs=5)  
;  
run;
```

# FREQ Procedure

- The FREQ procedure produces a one-way frequency table for each variable named in the TABLES statement.

```
PROC FREQ DATA=SAS-data-set;  
  <TABLES variable(s) </ options>>;  
RUN;
```

- Use options in the TABLES statement to suppress the display of selected default statistics.

Option	Description
NOCUM	Suppresses the cumulative statistics.
NOPERCENT	Suppresses the percentage display.

# FREQ procedure

```
proc freq data = demo.mydata nlevels ;
  tables gt treat / nopercent nocum ;
run;
```

- The *NLEVELS* option displays a table that provides the number of distinct values for each analysis variable.

Number of Variable Levels	
Variable	Levels
gt	5
treat	2

	Frequency	Percent	Cumulative Frequency	Cumulative Percent
erf115oe	117	20.00	117	20.00
luh-3	119	20.34	236	40.34
luh-4	119	20.34	355	60.68
seu-4	111	18.97	466	79.66
wt	119	20.34	585	100.00

# Crosstabulation Table

- An asterisk between two variables generates a two-way frequency table, or *crosstabulation table*.

```
proc freq data = demo.mydata nlevels ;
  tables gt * treat ;
run;
```

The diagram shows the PROC FREQ code with annotations. Brackets under 'gt' and 'treat' point to green boxes labeled 'rows' and 'columns' respectively, indicating that 'gt' represents the rows and 'treat' represents the columns in the crosstabulation table.

- A two-way frequency table generates a single table with statistics for each distinct combination of values of the selected variables.

# Viewing the Output

- PROC FREQ Output

gt	Table of gt by treat		
	control	npa	Total
erf115oe	53	64	117
	9.06	10.94	20.00
	45.30	54.70	
	18.93	20.98	
luh-3	60	59	119
	10.26	10.09	20.34
	50.42	49.58	
	21.43	19.34	
luh-4	57	62	119
	9.74	10.60	20.34
	47.90	52.10	
	20.36	20.33	
seu-4	54	57	111
	9.23	9.74	18.97
	48.65	51.35	
	19.29	18.69	
wt	56	63	119
	9.57	10.77	20.34
	47.06	52.94	
	20.00	20.66	
Total	280	305	585
	47.86	52.14	100.00

# Options to Suppress Statistics

- Use options in the TABLES statement to suppress the display of selected default statistics.

**TABLES variable(s) / options ;**

Option	Description
NOROW	Suppresses the display of the row percentage.
NOCOL	Suppresses the display of the column percentage.
NOPERCENT	Suppresses the percentage display.
NOFREQ	Suppresses the frequency display.

# LIST and CROSSLIST Options

gt	treat	Frequency	Percent	Cumulative Frequency	Cumulative Percent
erf115oe	control	53	9.06	53	9.06
erf115oe	npa	64	10.94	117	20.00
luh-3	control	60	10.26	177	30.26
luh-3	npa	59	10.09	236	40.34
luh-4	control	57	9.74	293	50.09
luh-4	npa	62	10.60	355	60.68
seu-4	control	54	9.23	409	69.91
seu-4	npa	57	9.74	466	79.66
wt	control	56	9.57	522	89.23
wt	npa	63	10.77	585	100.00

← LIST  
option

Table of gt by treat						
gt	treat	Frequency	Percent	Row Percent	Column Percent	
erf115oe	control	53	9.06	45.30	18.93	
	npa	64	10.94	54.70	20.98	
	Total	117	20.00	100.00		
luh-3	control	60	10.26	50.42	21.43	
	npa	59	10.09	49.58	19.34	
	Total	119	20.34	100.00		
luh-4	control	57	9.74	47.90	20.36	
	npa	62	10.60	52.10	20.33	
	Total	119	20.34	100.00		
seu-4	control	54	9.23	48.65	19.29	
	npa	57	9.74	51.35	18.69	
	Total	111	18.97	100.00		
wt	control	56	9.57	47.06	20.00	
	npa	63	10.77	52.94	20.66	
	Total	119	20.34	100.00		
Total	control	280	47.86		100.00	
	npa	305	52.14		100.00	
	Total	585	100.00			

→ CROSSLIST  
option

# PROCTABULATE

- Singular table

```
title 'Summary statistics by genotype';
proc tabulate data = demo.mydata ;
  class gt ;
  var amyloplastheight ;
  table gt, amyloplastheight ;
run;
```

rows      columns

Summary statistics by genotype

	amyloplastheight	Sum	
		gt	erf115oe
erf115oe			20559.00
luh-3			16426.00
luh-4			15081.00
seu-4			17428.00
wt			12596.00

# PROCTABULATE

- Nested tables

```
title 'Nested tables';
proc tabulate data = mydata ;
  class gt treat;
  var amyloplastheight ;
  table gt, treat * amyloplastheight ;
  table gt * treat, amyloplastheight ;
run;
```

Nested tables

	treat	
	control	npa
	amyloplastheight	amyloplastheight
	Sum	Sum
gt		
erf115oe	3541.00	17018.00
luh-3	3575.00	12851.00
luh-4	3429.00	11652.00
seu-4	3663.00	13765.00
wt	3255.00	9341.00

Nested table

		amyloplastheight
		Sum
gt	treat	
erf115oe	control	3541.00
	npa	17018.00
luh-3	control	3575.00
	npa	12851.00
luh-4	control	3429.00
	npa	11652.00
seu-4	control	3663.00
	npa	13765.00
wt	control	3255.00
	npa	9341.00

# PROCTABULATE

- Selecting statistics and formatting

```
title 'Selected statistics';
proc tabulate data = mydata format=8.2;
  class gt treat;
  var amyloplastheight ;
  table gt * treat, amyloplastheight*(N*f=3.0 Nmiss*f=3.0 mean lclm uclm) ;
  label gt= 'genotype';
  label treat= 'treatment';
run;
```

		Selected statistics				
		amyloplastheight				
genotype	treatment	N	NMiss	Mean	95_LCLM	95_UCLM
erf115oe	control	53	0	66.81	64.22	69.40
	npa	64	0	265.91	244.59	287.22
luh-3	control	60	0	59.58	57.78	61.39
	npa	59	0	217.81	191.08	244.54
luh-4	control	57	0	60.16	58.50	61.81
	npa	62	0	187.94	161.64	214.23
seu-4	control	54	0	67.83	65.65	70.02
	npa	57	0	241.49	223.20	259.78
wt	control	56	0	58.13	56.32	59.93
	npa	63	0	148.27	134.80	161.74

# Table output

```
ods rtf file = "C:\folder\subfolder\filename.rtf" style=analysis ;  
proc tabulate ...  
run;  
ods rtf close;
```

# Try it yourself

- Prepare some data
- Open SAS/EG and run the sascode **data\_import\_EG.sas** available on  
\\psb.ugent.be\shares\psb-department\psb\_abb\STATISTICS\SCRIPTS\SAS

# SAS online documentation

- <http://documentation.sas.com>
- tutorials: <https://video.sas.com/>
- <https://blogs.sas.com/content/>
- SASproceedings: <https://www.lexjansen.com/>
- YouTube channel: <https://www.youtube.com/SASsoftware>
- Asking for help:
  - <https://communities.sas.com/>
  - "SAS Technical Support" <[support@sbx.sas.com](mailto:support@sbx.sas.com)>;