

A Seamless Integration of Gretl and R

Sagiru Mati (PhD)

1 About gretlR

gretlR is an R package that can run `gretl` program from R Markdown.

2 Installation

gretlR can be installed using the following commands in R.

```
install.packages("gretlR")
```

OR

```
devtools::install_github('sagirumati/gretlR')
```

3 Usage

Please load the gretlR package as follows:

```
```${r gretlR}
library(gretlR)
```
```

Then create a chunk for `gretl` as shown below:

```
```${gretl gretlR,eval=T,echo=T,comment=NULL}
nulldata 500
set seed 13
gretl1 = normal()
gretl2 = normal()
setobs 12 1980:01 --time-series
gnuplot gretl1 --time-series --with-lines --output="line.png"
gnuplot gretl2 gretl1 --output="scatter.png"
ols gretl1 const gretl2
modeltab add
tabprint --output="olsTable.Rmd"
tabprint --output="olsTable.tex"
tabprint --output="olsTable.csv"
eqnprint --output="olsEquation.tex"
```
```

The above chunk creates a `gretl` program with the chunk's content, then automatically run the `gretl` script, which will save `gretl` outputs in the new folder `gretlR` created in the current working directory.

4 include_graph function

We can *dynamically and reproducibly* fetch the `gretl` graph object we created with the `gretl` chunk using the following R chunk:

For the scatter graph:

```
include_graph(chunk = "gretlR",graph = "scatter.png")
```



Figure 1: Scatter graph produced by `gretl` chunk

or the line graph:

```
include_graph(chunk = "gretlR",graph = "line.png")
```

5 include_tex function

we can also include the equation of the OLS generated by the `gretl` chunk and save as `olsEquation.tex`.

If the output is `pdf`, one can use the raw `LaTeX` codes as follows:

```
\input{gretlr/gretlR/olsEquation.tex}
```

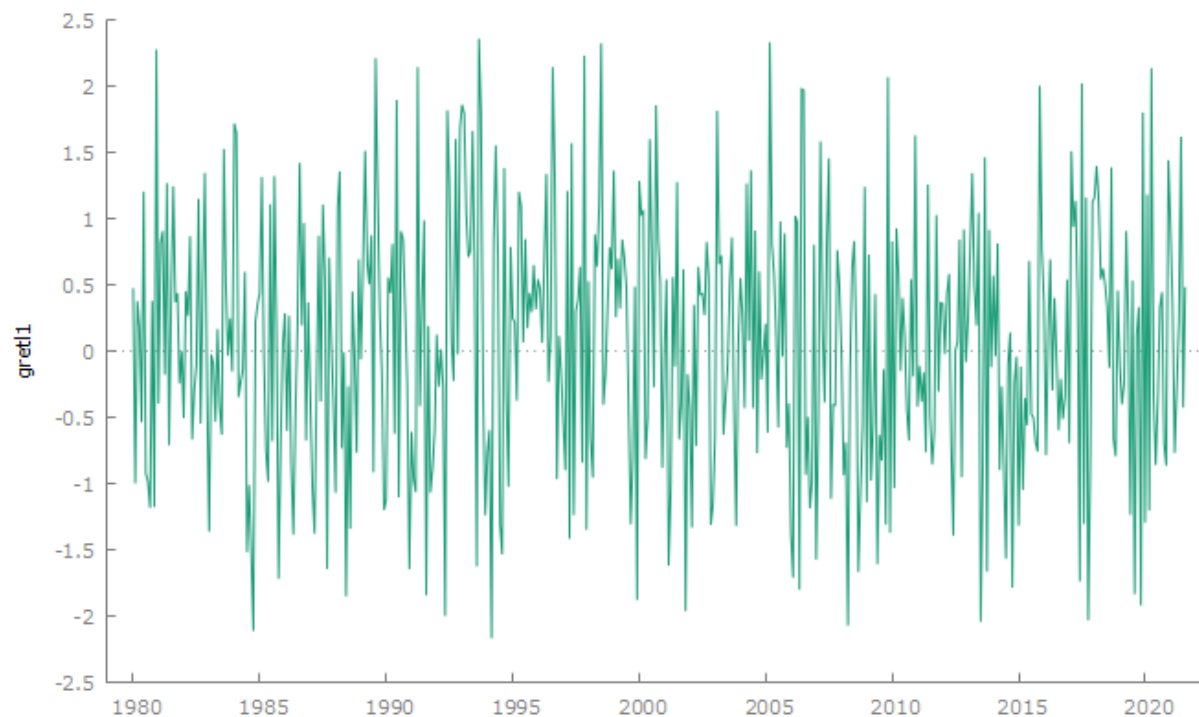


Figure 2: Line graph produced by gretl chunk

$$\widehat{\text{gretl1}} = 0.0610266 + 0.0239587 \text{gretl2}$$

(0.043179) (0.042056)

$$T = 500 \quad \bar{R}^2 = -0.0014 \quad F(1, 498) = 0.32454 \quad \hat{\sigma} = 0.96025$$

(standard errors in parentheses)

Or use `include_tex` function to include the equation as shown below:

```
include_tex(chunk = "gretlR", tex = "olsEquation")
```

$$\widehat{\text{gretl1}} = 0.0610266 + 0.0239587 \text{gretl2}$$

(0.043179) (0.042056)

$$T = 500 \quad \bar{R}^2 = -0.0014 \quad F(1, 498) = 0.32454 \quad \hat{\sigma} = 0.96025$$

(standard errors in parentheses)

```
include_tex(chunk = "gretlR", tex = "olsTable", start = 7, end = 24)
```

| | Coefficient | Std. Error | t-ratio | p-value |
|--------|-------------|------------|---------|---------|
| const | 0.0610266 | 0.0431785 | 1.413 | 0.1582 |
| gretl2 | 0.0239587 | 0.0420559 | 0.5697 | 0.5691 |

The OLS table output is saved by the `gretl` chunk as `olsTable.Rmd`. The entire OLS table output can be included as child document as follows:

```
```{r child, child='gretlr/gretlR/olsTable.Rmd'}
```

...

Model 1: OLS, using observations 1980:01–2021:08 ( $T = 500$ )  
Dependent variable: gretl1

	Coefficient	Std. Error	t-ratio	p-value
const	0.0610266	0.0431785	1.413	0.1582
gretl2	0.0239587	0.0420559	0.5697	0.5691
Mean dependent var	0.058464	S.D. dependent var	0.959598	
Sum squared resid	459.1937	S.E. of regression	0.960248	
$R^2$	0.000651	Adjusted $R^2$	-0.001355	
$F(1, 498)$	0.324542	P-value( $F$ )	0.569148	
Log-likelihood	-688.1853	Akaike criterion	1380.371	
Schwarz criterion	1388.800	Hannan–Quinn	1383.678	
$\hat{\rho}$	-0.046001	Durbin–Watson	2.091190	

## 6 import\_kable function

The `gretl` chunk also saves the OLS table as `olsTable.csv`. The `import_kable` function can be used to import it as a table. further customisation can be done with `kableExtra` package.

```
import_kable(chunk = "gretlR", file = "olsTable.csv", caption="Table generated from gretl
 chunk", start=3,end=7,digits=2) |>
kableExtra::kable_styling(latex_options = c("basic","hold_position","scale_down")) |>
kableExtra::row_spec(0,bold=T)
```

Table 1: Table generated from gretl chunk

	<b>coefficient</b>	<b>std. error</b>	<b>t-ratio</b>	<b>p-value</b>
const	0.06	0.04	1.41	0.16
gretl2	0.02	0.04	0.57	0.57

## 7 write\_inp function

This function writes `gretl` file.

```
code=r'(nulldata 500
set seed 13
gretl1 = normal()
gretl2 = normal()
setobs 12 1980:01 --time-series
gnuplot gretl1 --time-series --with-lines --output="line.png"
gnuplot gretl2 gretl1 --output="scatter.png"
)'
```

`write_inp(code,path="gretlCodes")`

## 8 exec\_inp function

This function executes existing gretl files.

```
code=r'(nulldata 500
set seed 13
gretl1 = normal()
gretl2 = normal()
setobs 12 1980:01 --time-series
gnuplot gretl1 --time-series --with-lines --output="line.png"
gnuplot gretl2 gretl1 --output="scatter.png"
) '
write_inp(code,path="SomeFolder/gretlCodes")
exec_inp("someFolder/gretlCodes")
```

## 9 exec\_gretl function

This function creates gretlfile from R object or a set of character strings and executes it. It is a combination of write\_inp and exec\_inp functions.

```
code=r'(nulldata 500
set seed 13
gretl1 = normal()
gretl2 = normal()
setobs 12 1980:01 --time-series
gnuplot gretl1 --time-series --with-lines --output="line.png"
gnuplot gretl2 gretl1 --output="scatter.png"
) '
exec_gretl(code)
```

Please visit my [Github](#) to download a set of example files.