# gretlR Manual: Version 1

Sagiru Mati

2020-06-14

## About gretlR

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

## Installation

gretlR can be installed using the following commands in R.

```
```{r}
install.packages("gretlR")
```
```

OR

```
```{r}
devtools::install_github('sagirumati/gretlR')
```
```

## Usage

Please load the gretlR package as follows:

```
```{r,eval=TRUE}
library(gretlR)
```
```

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

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.

## 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:

````
```{r,eval=TRUE}
#| label: fig-graph
#| fig-cap: Gretl graph
#| out-width: 90%
#| out-height: 90%
include_graph(chunk = "gretlR",graph = "scatter.png")
```
````

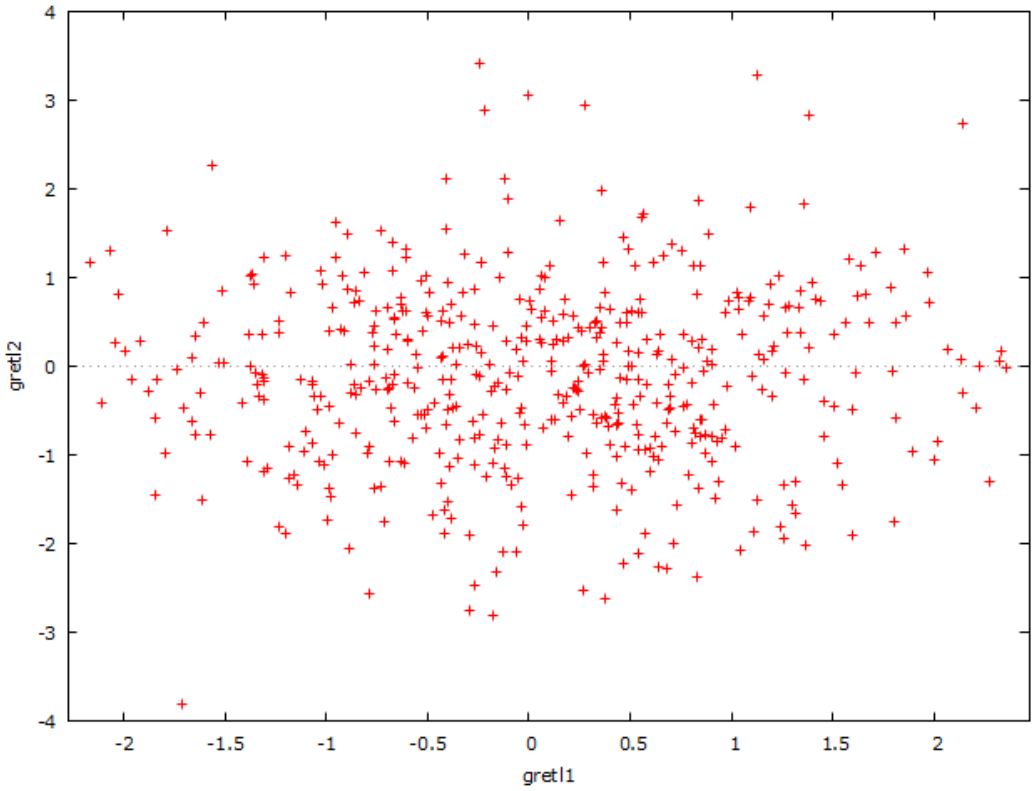


Figure 1: Gretl graph

or the line graph:

```
```{r,eval=TRUE}
#| label: fig-graph1
#| fig-cap: Another Gretl graph
#| out-width: 90%
#| out-height: 90%
include_graph(chunk = "gretlR",graph = "line.png")
```
```

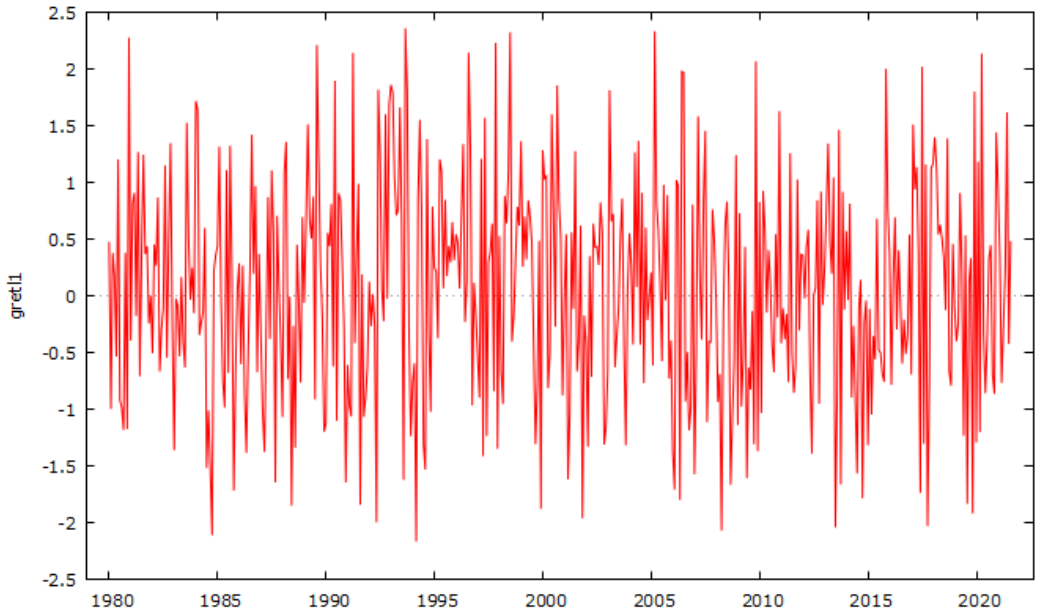


Figure 2: Another Gretl graph

## 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}`

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

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

Or include lines 7 to 24 of `olsTable.tex` produced by the gretl chunk:

```
```{r,eval=TRUE}
include_tex(chunk = "gretlR",tex = "olsTAble",start =8,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 included as child document as follows:

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 |

## import_kable function

The `gretl` chunk also saves the OSL 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.

```
```{r,eval=TRUE}
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

| | **coefficient** | **std. error** | **t-ratio** | **p-value** |
|---|---|---|---|---|
| const | 0.06 | 0.04 | 1.41 | 0.16 |
| gretl2 | 0.02 | 0.04 | 0.57 | 0.57 |

## 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")
```

## 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")
```

## exec_gretl function

This function creates `gretl`file 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)
```

## Demo

Demo can be accessed via `demo(package="gretlR")`.

```
demo(exec_inp)
demo(write_inp)
demo(exec_gretl)
```

## R Markdown template

The R Markdown template for the `gretlR` can be accessed via `file -> New File -> R Markdown -> From Template -> gretlR`

## Similar Packages

Similar packages include DynareR (Mati, 2020a, 2022a), EviewsR (Mati, 2020b, 2022b; Mati et al., 2023), and URooTab (Mati, 2023b, 2023a)

For further details, consult Mati (2020c) and Mati (2022c).

Please download a set of example files from Github.

## References


Mati, S. (2020a). DynareR: Bringing the power of dynare to R, R Markdown, and Quarto. *CRAN*. https://CRAN.R-project.org/package=DynareR

Mati, S. (2020b). *EviewsR: A seamless integration of EViews and R*. https://CRAN.R-project.org/package=EviewsR

Mati, S. (2020c). *gretlR: A seamless integration of Gretl and R*. https://CRAN.R-project.org/package=gretlR

Mati, S. (2022a). *Package "DynareR"*. https://cran.r-project.org/web/packages/DynareR/DynareR.pdf

Mati, S. (2022b). *Package "EviewsR"*. https://cran.r-project.org/web/packages/EviewsR/EviewsR.pdf

Mati, S. (2022c). *Package "gretlR"*. https://cran.r-project.org/web/packages/gretlR/gretlR.pdf

Mati, S. (2023a). *Package "URooTab"*. https://cran.r-project.org/web/packages/URooTab/URooTab.pdf

Mati, S. (2023b). *URooTab: Tabular reporting of EViews unit root tests*. https://github.com/sagirumati/URooTab

Mati, S., Civcir, I., & Abba, S. I. (2023). EviewsR: An r package for dynamic and reproducible research using EViews, r, r markdown and quarto. *The R Journal*, *15*(2), 169–205. https://doi.org/10.32614/rj-2023-045