URooTab: Tabular Reporting of EViews Unit Root Tests

Sagiru Mati

2023-08-30

Please do not forget to cite the package as follows:

Plain text:

Mati S. (2023). URooTab: Tabular Reporting of EViews Unit Root Tests. CRAN, https://github.com/sagirumati/URooTab

Bibtex:

```
@Manual{Mati2023,
  title = {{URooTab}: Tabular Reporting of {EViews} Unit Root Tests},
  author = {Sagiru Mati},
  publisher = {CRAN},
  url = {https://github.com/sagirumati/URooTab},
}
```

About the Author

The author of this package, **Sagiru Mati**, obtained his PhD in Economics from the Near East University, North Cyprus. He works at the Department of Economics, Yusuf Maitama Sule (Northwest) University, Kano, Nigeria. Please visit his website for more details.

Please follow his publications on:

Google Scholar

ResearchGate

Web of Science

ORCID: 0000-0003-1413-3974

About URooTab

URooTab is an R package that can conducts `EViews` unit root tests and report them in tabular form.

Why URooTab?

While there are R packages and EViews add-ins available for presenting unit root tests in tabular form, none of them incorporates **EViews** procedures within the R environment. Specifically:

- I wish I could conduct unit root using EViews routines in R, R Markdown or Quarto document
- I wish I could dynamically import the results of the unit root tests individually or at once into R, R Markdown or Quarto document without switching between these applications back and forth.
- I wish I could use an R function to report unit root test in a table style suitable for publication.
- I wish I could automatically format the table in `Latex`, `html`, `pandoc` and `markdown`.
- I wish I could do all of the above from R, R Markdown or Quarto without opening the EViews!!!

Installation

URooTab can be installed using the following commands in R.

~~~
```{r installation,eval=F}
install.packages("URooTab")

            OR

devtools::install_github('sagirumati/URooTab')
```
~~~

Setup

To run the package successfully, you need to do one of the following

- Add EViews installation folder to path (**Environment Variables**).
- Don't do anything if the name of EViews executable is one of the following: `EViews13_x64`, `EViews13_x86`, `EViews12_x64`, `EViews12_x86`, `EViews11_x64`, `EViews11_x86`, `EViews10_x64`, `EViews10_x86`, `EViews9_x64`, `EViews9_x86`, `EViews10`. The package will find the executable automatically.
- Rename the Eviews executable to `eviews` or one of the names above.
- Alternatively, you can use `set_eviews_path()` function to set the path the EViews executable as follows:

~~~
```{r eval=F}
library(EviewR)
set_eviews_path("C:/Program Files (x86)/EViews 10/EViews10.exe")
```
~~~

Usage

Please load the URooTab package as follows:

~~~
```{r}
library(URooTab)
```
~~~

Ways to use URooTab

The package can work with base R, R Markdown or Quarto document.

URooTab along with R Markdown or Quarto document

You can use `URooTab` in an R chunk in R Markdown or Quarto document:

The `uroot()` function reports all the available test (ADF and PP) at once. It is more suitable for Quarto document, which has both `tbl-cap` and `tbl-subcap` chunk options.

To produce Table @ref(tab:URooTab), use the R chunk below:

Notice the chunk option `results: asis` because `uroot()` is designed to print all the tables (ADF and PP) in the chunk. If you are producing multiple `kable` tables, `results: asis` is necessary. You can also use `kableExtra` package to further customise the table.

~~~
```{r}
#| label: URooTab
#| eval: true
#| results: asis

library(URooTab)
set.seed(1234) # for reproducibility
x=rnorm(100)
y=cumsum(x)
z=cumsum(y)

dataFrame=data.frame(x,y,z)
uroot(dataFrame, caption = "Unit Root Tests for x, y and Z")
```
~~~

Table 1: Unit Root Tests for x, y and Z

Variables	None	Constant	Constant and trend	None	Constant	Constant and trend	Decision
X	-8.300***	-8.396***	-8.815***	-8.274***	-8.239***	-8.214***	I(0)
Y	0.417	-1.907	0.026	-8.148***	-8.259***	-8.721***	I(1)
Z	-2.379**	-2.084	-2.938	0.417	-2.013	-0.033	I(2)

Table 2: Unit Root Tests for x, y and Z

Variables	None	Constant	Constant and trend	None	Constant	Constant and trend	Decision
X	-8.327***	-8.418***	-8.815***	-42.502***	-51.961***	-74.206***	I(0)
Y	0.275	-1.857	-0.066	-8.170***	-8.275***	-8.721***	I(1)
Z	6.659	3.450	-3.516**	0.274	-1.956	-0.109	I(2)

Variables	None	Constant	Constant and trend	None	Constant	Constant and trend	Decision

In R Markdown or Quarto document, `URooTab` is smart enough to recognise the document format and select the suitable table format.

URooTab along with base R.

In base R, you can get the table printed in console in the format you specify by the `format` argument.

We can create a dataframe as follows:

```r
library(URooTab)
set.seed(1234) # for reproducibility
x=rnorm(100)
y=cumsum(x)
z=cumsum(y)

dataFrame=data.frame(x,y,z)
```

The adf() function

To print ADF test results in `latex` format:

~~~
```{r}
#| label: adf
#| eval: false

adf(dataFrame,format = "latex",info="aic",
    caption = "ADF Unit Root Tests for x, y and Z")
```
~~~

Or

~~~
```{r}
#| label: adf1
#| eval: false
~~~

```
uroot(dataFrame,format = "latex",test = "adf",info="aic",
      caption = "ADF Unit Root Tests for x, y and Z")
```
```

The above code produces the following latex code:

```
\begin{table}[h]

\caption{ADF Unit Root Tests for x, y and Z}
\centering
\begin{tabular}[t]{llllllll}
\toprule
Variables & None & Constant & Constant and trend & None & Constant & Constant and trend & De
\midrule
X & -8.300*** & -8.396*** & -8.815*** & -7.494*** & -7.460*** & -7.445*** & I(0)\\
Y & 0.224 & -1.934 & 0.026 & -8.148*** & -8.259*** & -8.721*** & I(1)\\
Z & -2.379** & -2.084 & -2.938 & 0.233 & -2.221 & -0.033 & I(2)\\
\bottomrule
\end{tabular}
\end{table}
```

The pp() function

To print PP test results in `html` format:

````
```{r}
#| label: pp
#| eval: false

pp(dataFrame,format = "html",info="aic",caption = "PP Unit Root Tests for x, y and Z")
```
````

Or

````
```{r}
#| label: pp1
#| eval: false

uroot(dataFrame,format = "html",info="aic",test = "pp",
      caption = "PP Unit Root Tests for x, y and Z")
```
````
````

The above code produces the following `html` codes in console:

```
<table>
<caption>PP Unit Root Tests for x, y and Z</caption>
 <thead>
  <tr>
   <th style="text-align:left;"> Variables </th>
   <th style="text-align:left;"> None </th>
   <th style="text-align:left;"> Constant </th>
   <th style="text-align:left;"> Constant and trend </th>
   <th style="text-align:left;"> None </th>
   <th style="text-align:left;"> Constant </th>
   <th style="text-align:left;"> Constant and trend </th>
   <th style="text-align:left;"> Decision </th>
  </tr>
 </thead>
<tbody>
  <tr>
   <td style="text-align:left;"> X </td>
   <td style="text-align:left;"> -8.327*** </td>
   <td style="text-align:left;"> -8.418*** </td>
   <td style="text-align:left;"> -8.815*** </td>
   <td style="text-align:left;"> -42.502*** </td>
   <td style="text-align:left;"> -51.961*** </td>
   <td style="text-align:left;"> -74.206*** </td>
   <td style="text-align:left;"> I(0) </td>
  </tr>
  <tr>
   <td style="text-align:left;"> Y </td>
   <td style="text-align:left;"> 0.275 </td>
   <td style="text-align:left;"> -1.857 </td>
   <td style="text-align:left;"> -0.066 </td>
   <td style="text-align:left;"> -8.170*** </td>
   <td style="text-align:left;"> -8.275*** </td>
   <td style="text-align:left;"> -8.721*** </td>
   <td style="text-align:left;"> I(1) </td>
  </tr>
  <tr>
   <td style="text-align:left;"> Z </td>
   <td style="text-align:left;"> 6.659 </td>
   <td style="text-align:left;"> 3.450 </td>
   <td style="text-align:left;"> -3.516** </td>
   <td style="text-align:left;"> 0.274 </td>
   <td style="text-align:left;"> -1.956 </td>
   <td style="text-align:left;"> -0.109 </td>
   <td style="text-align:left;"> I(2) </td>
  </tr>
</tbody>
</table>
```

### The uroot() function

The `uroot()` function is a generic function that can be used to conduct any unit root test. Setting `test="adf"` conducts ADF test, while `test="pp"` conducts PP test. If `test` argument is not specified, the `uroot()` function conducts all the test at once.