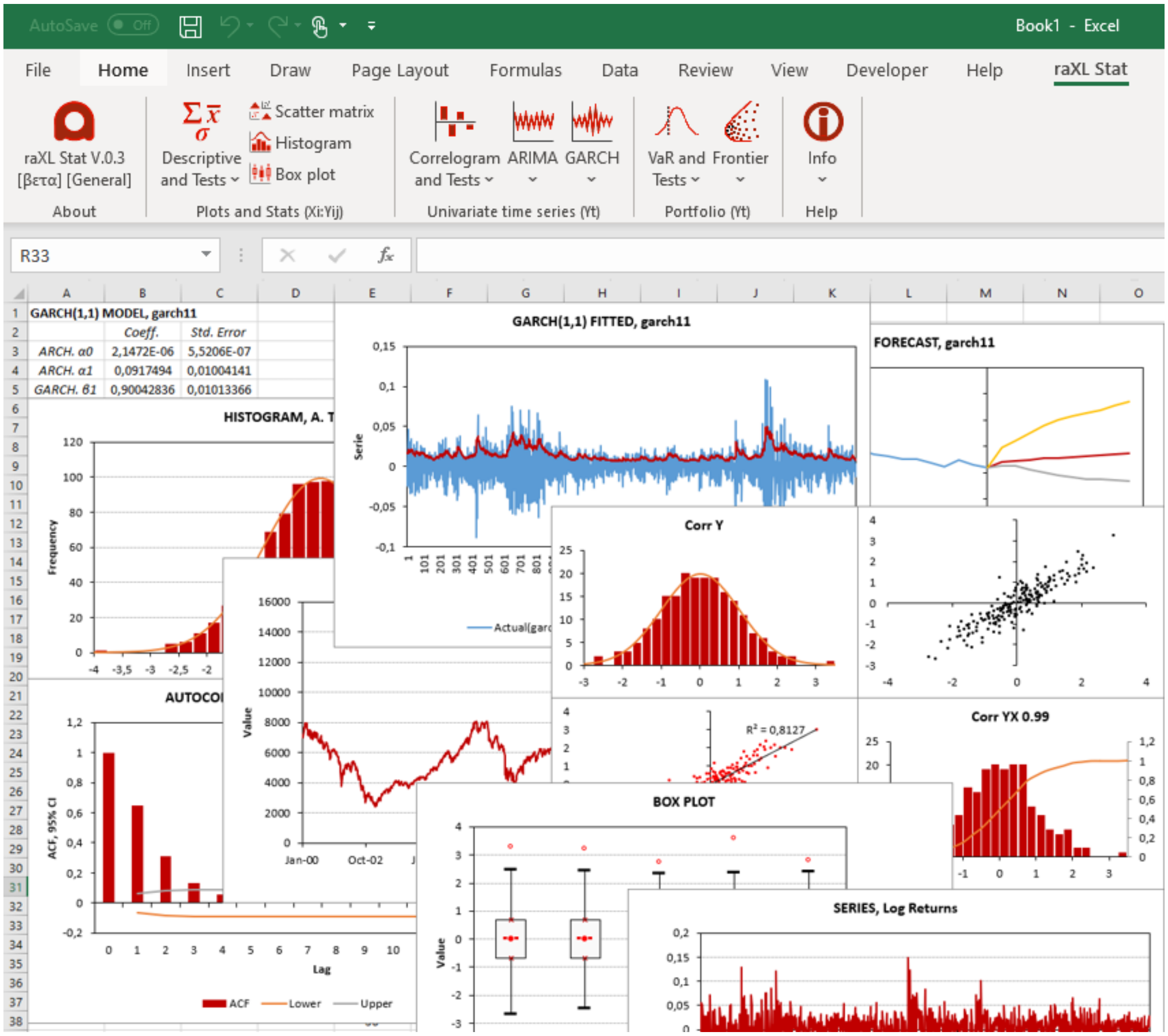
raXL Stat V.0.3 [βετα]

Statistical Add-in for Data Science in Excel

User manual



	Coeff.	Std. Error
GARCH(1,1) MODEL, garch11		
ARCH. α0	2,1472E-06	5,5206E-07
ARCH. α1	0,0917494	0,01004141
GARCH. β1	0,90042836	0,01013366

La Paz, November 2024

Doc. Version V.0.3

Content

Preface and Disclaimer

raXL Stat is a statistical add-in for Data Science for Microsoft Excel[1].

This Getting Started Guide is intended to guide you through the process of using raXL Stat to give you an overview of the menu and functions of raXL Stat. The information contained herein is not error-free and is subject to change without notice. If you find any errors, please let us know.

The raXL Stat software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Your access to and use of this material is subject to the terms and conditions of the End User License Agreement, which you agree to abide by.

The software is intended for general use in information management applications. It is not intended for use in inherently hazardous or potentially hazardous applications. If you intend to use this software in hazardous applications, you must take all appropriate precautionary measures to ensure the safe use of this software.

This manual is not distributed under a GPL license. You may create a printed copy of this manual for your personal use only. You will not publish or distribute this manual in any form or on any media without prior written consent. Conversion to other formats is permitted as long as the actual content is not altered or edited.

For more information, please visit https://ruben-apaza.blogspot.com/p/raxl-stat.html or you can contact rubenfapaza@gmail.com.

[1] Microsoft, Windows, Excel, and the Office logo are registered trademarks of Microsoft Corporation.

LICENSE AGREEMENT

raXL Management (hereinafter "Administration") grants client’s permission to use this plugin in accordance with the software terms of use below.

Article 1 Disclaimer

(1) The Administration does not offer any warranty or compensation regarding this software. Therefore, any problems arising with this plugin will be handled at the customer's own expense and responsibility.

(2) If you only use the Trial version functions of this software, technical support may not be provided.

Article 2 Copyright

This software is protected by copyright laws, international copyright agreements, and other intellectual property protection laws and agreements. You may not use the Software for commercial purposes of time-sharing, rental, leasing, or operating a service bureau, whether paid or free of charge. You may not distribute, sublicense, market, or sell this software to any third party. You may not assign, transfer, or delegate your rights or obligations under this License to any third party.

However, the Customer may transfer the Software to a third party on condition that the third party complies with the terms of use of this Software under the Customer's responsibility and that the Customer waives the right to use the Software.

Article 3 Termination of the contract

(1) If the Customer is unable to comply with all provisions contained in these Terms of Use, the contract based on these Terms of Use shall be terminated immediately without notice from the Administration.

(2) In the case of the preceding paragraph, the Administration will not refund the usage fee to the customer.

1. Introduction

raXL Stat is an add-in for Microsoft Excel on Windows that turns your spreadsheet into a quantitative and predictive analysis software, offering a collection of functions for creating statistical, econometric, financial and mathematical models. You can call these functions directly from your spreadsheet and they will return results directly to it.

raXL Stat is a statistical analysis software that will offer[2] easy-to-use tools to perform and deliver quality work in a short time. It is developed[3] for use by both beginners and experts. The easiest and most intuitive way to execute the functions is through the Excel ribbon menu. If necessary, the user can directly write the functions in the spreadsheet cells or can invoke the functions from VBA (Visual Basic for Application) programming.

A feature of raXL Stat is that it requires no installation (portable .xll file), it integrates seamlessly with Excel with new User Defined Functions (UDF), automated charting, a rich set of shortcuts, and an intuitive user interface for use. It integrates with the version of Excel you have installed on your Personal Computer (PC), not the Windows version.

With the raXL Stat plugin v.0.3 [Beta] you can do the following:

- Calculate and graph the Autocorrelation Function (ACF).
- Calculate and graph the Partial Autocorrelation Function (PACF).
- Perform the White Noise test and independence with the Ljung-Box or Box-Pierce test.
- Perform the Unit Root and Stationarity test with the Augmented Dickey-Fuller (ADF) test or Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test.
- Calculate the coefficients, estimate, forecast and graph the ARIMA(p,d,q) models, that is, AR(p), MA(q) and ARMA(p,q).
- Calculate the coefficients, estimate, forecast and graph the ARCH(p) and GARCH(p,q) models.
- Plot Histograms with cumulative curve or normal curve.
- Plot the box plot with outliers.
- Perform the Normality test with the Shapiro-Wilk, Anderson-Darling and Jarque-Bera tests.
- Make a table with descriptive statistics with a normality test.
- Make a table with a Covariance matrix or a Correlation Coefficient matrix with options for determination coefficients (R^2) or proof that there is no correlation (R=0).
- Count missing, blank, or missing data in a data range.
- Make the matrix dispersion graph.
- Calculate and graph the Value at Risk (VaR) of an investment portfolio.
- Perform Backtesting with z and LR-Kupiec test.
- Calculate the weights, risk and return of an investment portfolio.
- Calculate the minimum variance and tangency weights of an investment portfolio.

[2] raXL Stat version v.0 [Beta] is a test version to which new public functions will be added to the existing 30.

[3] Acknowledgment: raXL Stat uses Excel-DNA: Copyright (c) 2024 Govert van Drimmelen.

- Calculate and graph the Efficient Portfolio Frontier (EPF) and the Capital Market Line (CML) of an investment portfolio.
- Run a point simulation of an investment portfolio for the Efficient Frontier.
- Version 0.3 has over 76 additional public User Defined Functions (UPFs)[4].

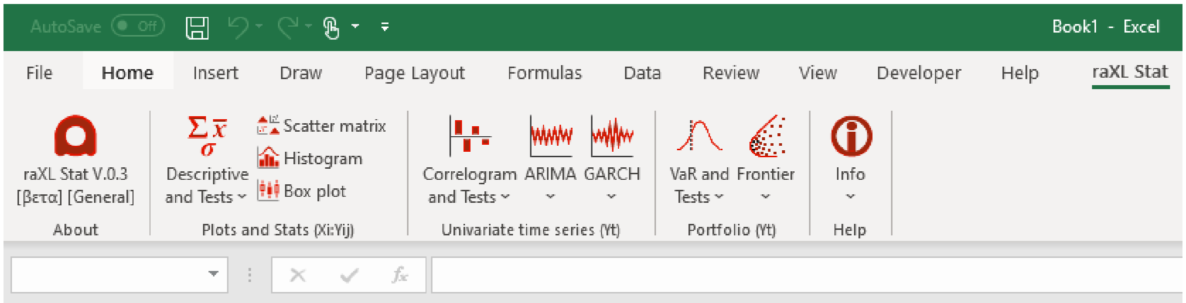


Figure 1

2. Download and use

In summary, since the add-in does not require installation, you just have to download the raXL_Stat-v0.3.zip file[5](https://ruben-apaza.blogspot.com/p/raxl-stat.html) Trial or Licensed, unzip the .zip and open the .xll add-in (Figure 2) depending on the Excel 32 or 64 bit and click on "Enable this add-in for this session only" (Figure 4). If it does not open or does not appear in the Excel ribbon menu (Figure 1), unlock it by right-clicking on the .xll add-in in properties and there check unlock and accept, and again open the .xll and it will appear in the Excel ribbon menu “raXL Stat” for use.

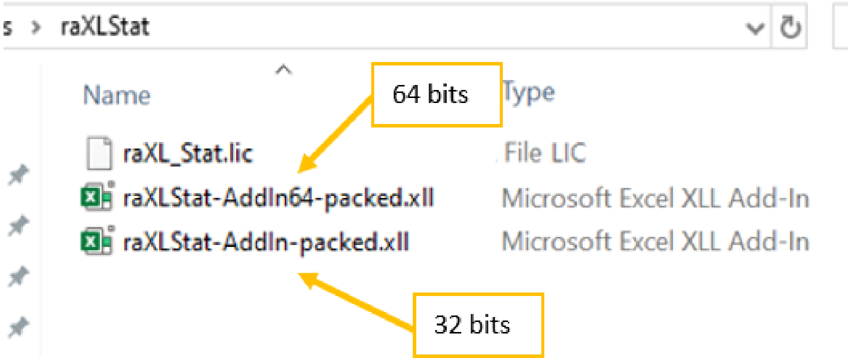
raXL Stat Technical Details

Before you start downloading raXL Stat for Excel, make sure that the system specifications listed below are available

- Setup File Name: raXL_Stat-v0.3.zip
- Setup Size: 4.0 MB
- Setup Type: Offline Installer / Full Standalone Setup
- Mechanical Compatibility: 32-bit (x86) / 64-bit (x64)
- Latest Version Released: November 15, 2024

[4] The ARIMA and GARCH functions use the Maximum Likelihood Estimation (MLE) method together with the Newton-Raphson (NR) optimization algorithm, however, other optimization methods such as Levenberg-Marquardt, BHHH, BFGS and others under development will be added.

[5] The file contains: " raXLStat-AddIn-packed.xll " and " raXLStat-AddIn64-packed.xll ", if you have purchased the license it will include the " raXL_Stat.lic " license, in the Trial version it includes a " Getting Started " manual.



Name	Type
raXL_Stat.lic	File LIC
raXLStat-AddIn64-packed.xll	Microsoft Excel XLL Add-In
raXLStat-AddIn-packed.xll	Microsoft Excel XLL Add-In

Figure 2

However, below are step-by-step instructions on how to download and use the plugin.

2.1 License and 30-day trial

The raXL_Stat.lic license is provided in the raXL_Stat-v0.3.zip file upon purchase (Figure 2) this .lic file must go together with the .xll plugin, it is annual and generic, if you have any problems with the license, want to customize or buy only the raXL_Stat.lic license you can contact rubenfapaza@gmail.com. If you purchased only the license you must copy or extract it to the folder where the .xll plugin is located. The license in the Trial or 30-day Trial version is automatically activated.

- Microsoft Excel: Excel 2010, 2013, 2016, 2019, 2021, 2024, or Office 365
- OS: Windows 7/8/8.1/10/11
- RAM: 512 MB
- Hard Drive: 10 MB
- Processor: Intel Dual Core Processor or later
- Microsoft .NET: .NET Framework 4.5.2 or later

2.2. System Requirements

raXL Stat is an add-in that integrates with Excel installed on Window 7 or later, it is not compatible with other operating systems such as Mac, Linux and others. Therefore, before using, you need to have a working version of Microsoft Excel on your Windows PC. It requires Microsoft Excel to be installed, at least Excel 2010 version. It works on both 32-bit and 64-bit Excel systems. In addition to Excel, the "Microsoft .NET Framework 4.5.2" or later is required, installed by default on Windows, but can be obtained for free via the official Microsoft website[6].

2.3. Excel version

raXL Stat must be used for the version of Excel you have installed on your Personal Computer (PC), not the Windows version. Please refer to the instructions below if you are unsure whether you have the 32-bit or 64-bit version of Excel installed.

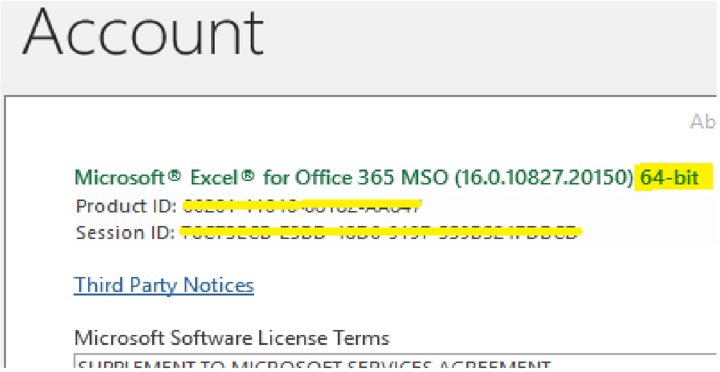
[6] The raXL Stat v.0[Beta] version is developed on .NET Framework 4.7.2, which and other versions can be downloaded at https://dotnet.microsoft.com/en-us/download/dotnet-framework/net472

The method to know if you have 32-bit or 64-bit Excel depends on the version of Excel you are using:

- Excel 2010: Select File, then Help. On the right side, look for "About Microsoft Excel." Just below that, you'll see the version and in parentheses, "32-bit" or "64-bit."
- Office 365 or Excel 2021, 2019, 2016, 2013: Follow the step-by-step instructions below.

If you already know which version of Excel you have, get the .zip file to use. Otherwise, follow the instructions below to find out which version of Excel you have installed:

1. Click the **File tab** in open Excel.
2. Then click on **Account** which is located in the bottom left corner.
3. On the right side of the window, you should see an **About Excel button.**
4. Click here and you should see, on the top line, the 32-bit or 64-bit version of Excel.



Account

Ab

Microsoft® Excel® for Office 365 MSO (16.0.10827.20150) 64-bit
Product ID: [redacted]
Session ID: [redacted]

Third Party Notices

Microsoft Software License Terms

SUPPLEMENT TO MICROSOFT SERVICES AGREEMENT

Figure 3

5. You can now run the appropriate 32-bit or 64-bit version of raXL Stat (Figure 2).

2.4. Download and use the plugin

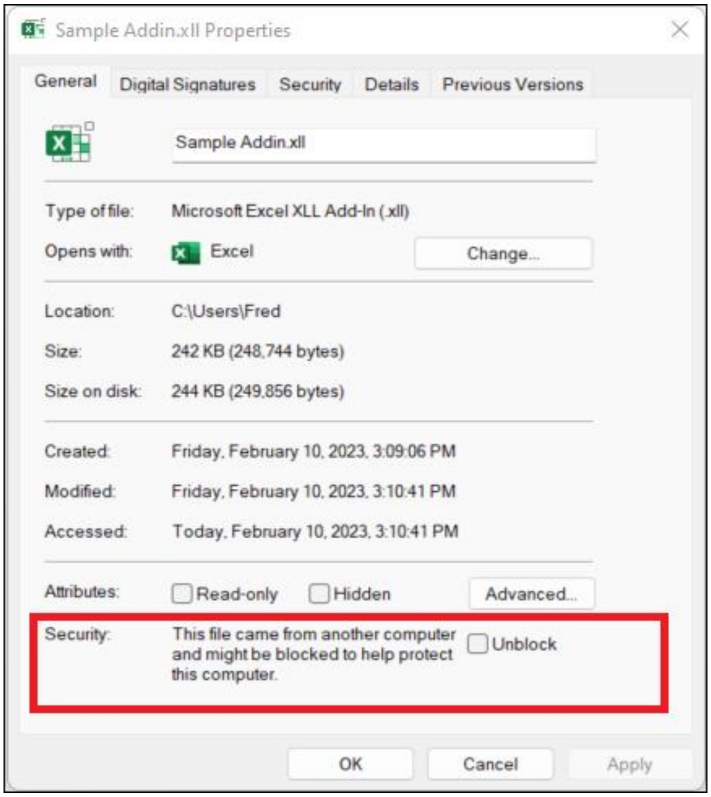
To use raXL Stat, here's how to download and use the plugin:

1. Download the file raXL_Stat-v0.3.zip (https://ruben-apaza.blogspot.com/p/raxl-stat.html) Trial or licensed.

2. Unzip the downloaded .zip file to a visible location, such as your desktop or wherever you can easily locate it. In the unzipped folder (Figure 2) you will find two .xll add-ons for 32-bit or 64-bit versions.

3. In the unzipped folder, unlock (Unblock) the .xll add-on from the Windows explorer[7], that is, right-click on the add-on and go to properties (Figure 4) and from there check the unblock check and then Apply (Apply) and Accept (Ok).

[7] As with any Excel file with Macros downloaded from the Internet such as .xlsx .xlsm .xlam, Microsoft Excel blocks .xll add-ins downloaded from the Internet by default.



Sample Addin.xll Properties

General | Digital Signatures | Security | Details | Previous Versions

Sample Addin.xll

Type of file: Microsoft Excel XLL Add-In (.xll)

Opens with: Excel

Location: C:\Users\Fred

Size: 242 KB (248,744 bytes)

Size on disk: 244 KB (249,856 bytes)

Created: Friday, February 10, 2023, 3:09:06 PM

Modified: Friday, February 10, 2023, 3:10:41 PM

Accessed: Today, February 10, 2023, 3:10:41 PM

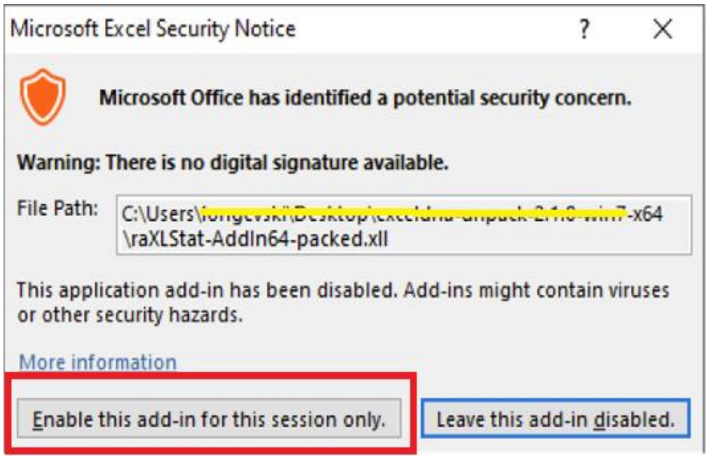
Attributes: □ Read-only □ Hidden

Security: This file came from another computer and might be blocked to help protect this computer. □ Unblock

Figure 4

4. Once unlocked, open the add-in " raXLStat-AddIn-packed.xll " or " raXLStat-AddIn64-packed.xll " (Figure 2) depending on the 32-bit or 64-bit Excel installed on your PC, by double-clicking on it or through Excel's File and Open menu.

5. When you open it, the "Microsoft Excel Security Notice" window will appear and you must click on "Enable this add-in for this session only" (Figure 5).



Microsoft Excel Security Notice

Microsoft Office has identified a potential security concern.

Warning: There is no digital signature available.

File Path: C:\Users\[redacted]-x64\raXLStat-AddIn64-packed.xll

This application add-in has been disabled. Add-ins might contain viruses or other security hazards.

?
More information
Enable this add-in for this session only.
Leave this add-in disabled.

Figure 5

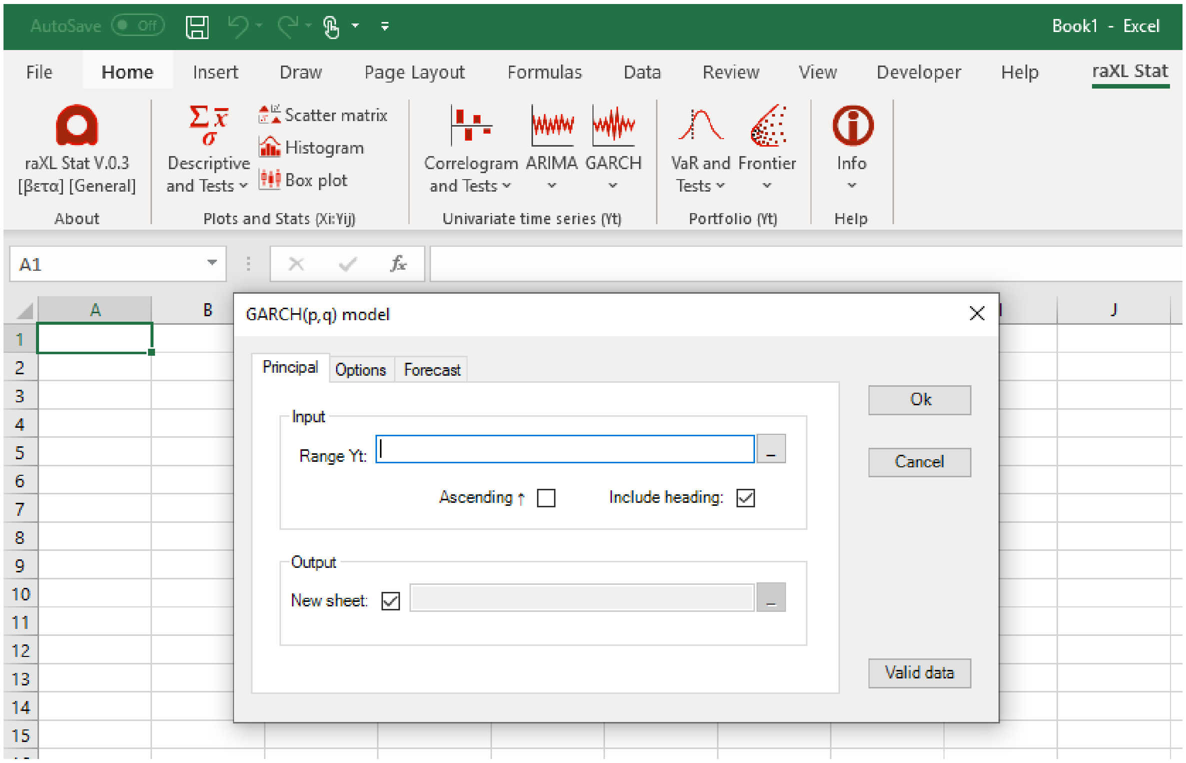
6. Once you have completed the steps you will have a new Menu in the Excel Ribbon called “raXL Stat” (Figure 1).

3. Functionalities

raXL Stat functions can be used and applied in three different ways, for more details you can check and watch the list of videos about using raXL Stat on our YouTube channel: https://www.youtube.com/watch?v=wYdGCkdN6cE&list=PLu4ltjreHhzO-cV1rHIis-K5_8numRqQV&pp=gAQBiAQB.

3.1. From the Menu

From the Excel ribbon menu of raXL Stat, it is the easiest way to execute the Functions. Select the desired operation and then fill in the menu form. When the Ok button is pressed, the syntax check is executed before calling the actual functions contained in the .xll add-in (Figure 2). In addition to the Main one, you have additional Options.



Range Yt:

Ascending ↑ □ Include heading: ☑

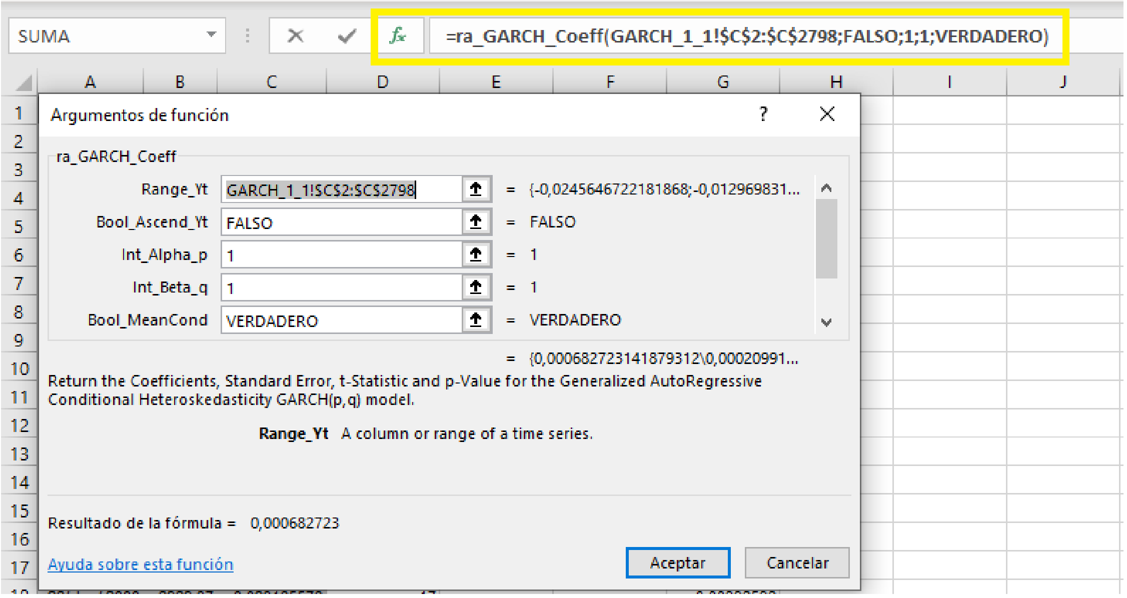
New sheet: ☑

Figure 6

3.2. Insert function

Typing functions manually into spreadsheet cells: In this case, the user must make sure to follow the correct syntax. Incorrect syntax will cause incorrect results, "#NUM!", Excel errors, or simply zero values "0". Also, it is very important to remember that in many cases the returned value of a function can be a vector or a matrix. In those cases, the function must be executed as a multi-cell array formula by pressing the CTRL+SHIFT+ENTER key combination, however, for Excel 365 or 2024 it is not required because it has dynamic array execution.

For more details see section 4. List of functions



=ra_GARCH_Coeff(GARCH_1_1!$C$2:$C$2798;FALSO;1;1;VERDADERO)

Argumentos de función

ra_GARCH_Coeff

Range_Yt GARCH_1_1!$C$2:$C$2798 = {-0,0245646722181868;-0,012969831...

Bool_Ascend_Yt FALSO = FALSO

Int_Alpha_p 1 = 1

Int_Beta_q 1 = 1

Bool_MeanCond VERDADERO = VERDADERO

= {0,000682723141879312\0,00020991...

Return the Coefficients, Standard Error, t-Statistic and p-Value for the Generalized AutoRegressive Conditional Heteroskedasticity GARCH(p,q) model.

Range_Yt A column or range of a time series.

Resultado de la fórmula = 0,000682723

Ayuda sobre esta función

Figure 7

3.3. Function from VBA

Called from an Excel VBA Macro, use the VBA Application.Run method.

```
Option Explicit
Option Base 1
Function Func_GARCHCoeff () As Variant
Application.Calculation = xlCalculationManual
Dim runResult As Variant
Dim result() As Double
Dim rngRange As Range
Dim boolAscend As Boolean
Dim intP As Integer
Dim intQ As Integer
Dim intMuCond As Boolean
Dim i As Integer
Set rngRange = ActiveSheet.Range("B2:B1001")
boolAscend = False
intP = 1
intQ = 1
intMuCond = False
ReDim result(intP + intQ + 1, 1)
runResult = Application.Run("ra_GARCH_Coeff", rngRange, boolAscend, intP, intQ, intMuCond)
For i = 1 To intP + intQ + 1
result(i, 1) = runResult(i, 1)
Next i
Func_GARCHCoeff = result
Application.Calculation = xlCalculationAutomatic
End Function
```

Figure 8

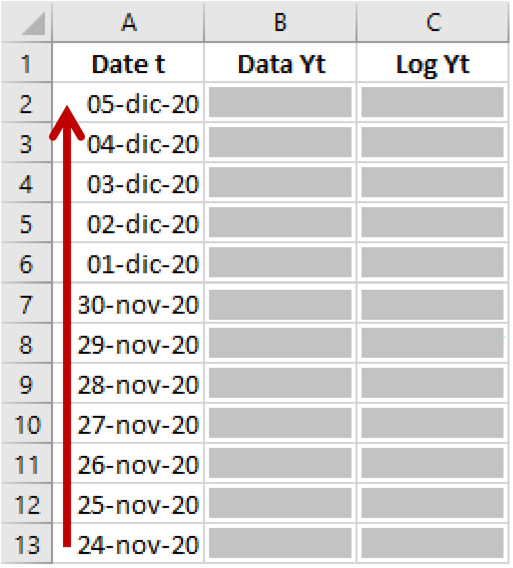
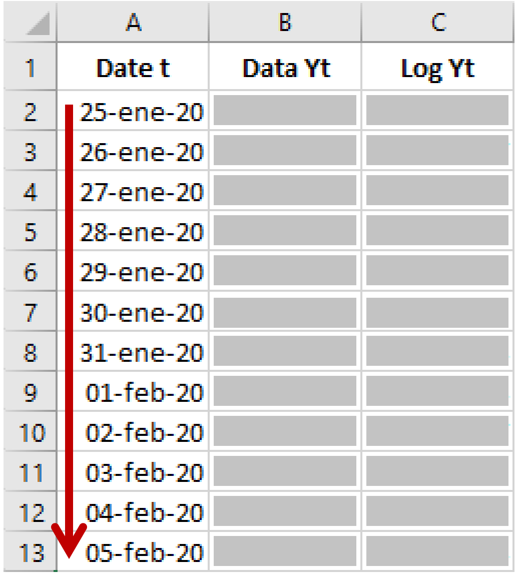
A Macro can be recorded with raXL Stat functions or run from the VBA editor as another UDF or Macro function by programming, for example (Figure 8) by creating a new "Function" called "RunGARCHCoeff" with the raXL Stat function "ra_GARCH_Coeff" and run from an Excel cell or by creating a "Sub" Macro to run with a Button from Excel. You can also see to use A. VBA Macro with Button.

4. List of functions

On the following pages, a list of raXL Stat User Defined Functions (UDFs) is provided, most of the functions can be shown as an example through the Excel ribbon menu. The function is called by typing in a cell "=" or by pressing (*fx*) in " insert function " (Figure 7) or by dropping down in "raXL Stat Add-in" to fill in the necessary arguments of the function.

4.1. Data preparation

Generally, when modeling data, for example, in time series, it is most common to show dates and values in columns (Ascending-Descending) in the same Excel sheet. Although the time or date is not fed into the model, it gives us a general idea about the chronological order of the data.

Ascending (↑)	**Descending (↓)**
It means that the data is in ascending order, that is, with the most recent date at the top of the sheet (The first value corresponds to the last or recent observation).	It means that the data is in Descending order, that is, with the most recent date at the bottom of the sheet (The first value corresponds to the first or initial observation).
	

	A	B	C
1	**Date t**	**Data Yt**	**Log Yt**
2	05-dic-20		
3	04-dic-20		
4	03-dic-20		
5	02-dic-20		
6	01-dic-20		
7	30-nov-20		
8	29-nov-20		
9	28-nov-20		
10	27-nov-20		
11	26-nov-20		
12	25-nov-20		
13	24-nov-20		

	A	B	C
1	**Date t**	**Data Yt**	**Log Yt**
2	25-ene-20		
3	26-ene-20		
4	27-ene-20		
5	28-ene-20		
6	29-ene-20		
7	30-ene-20		
8	31-ene-20		
9	01-feb-20		
10	02-feb-20		
11	03-feb-20		
12	04-feb-20		
13	05-feb-20		

To flip the data in the sheet you can use the ra_Range_Flip() function

4.2. UDF Functions

User Defined Functions (UDF) can be a value, a vector, or a matrix. In the case of a vector or matrix, the function is executed by pressing the CTRL+SHIFT+ENTER key combination, however, for

Excel Microsoft 365 or Excel 2024 it is not required because it works by executing dynamic matrices.

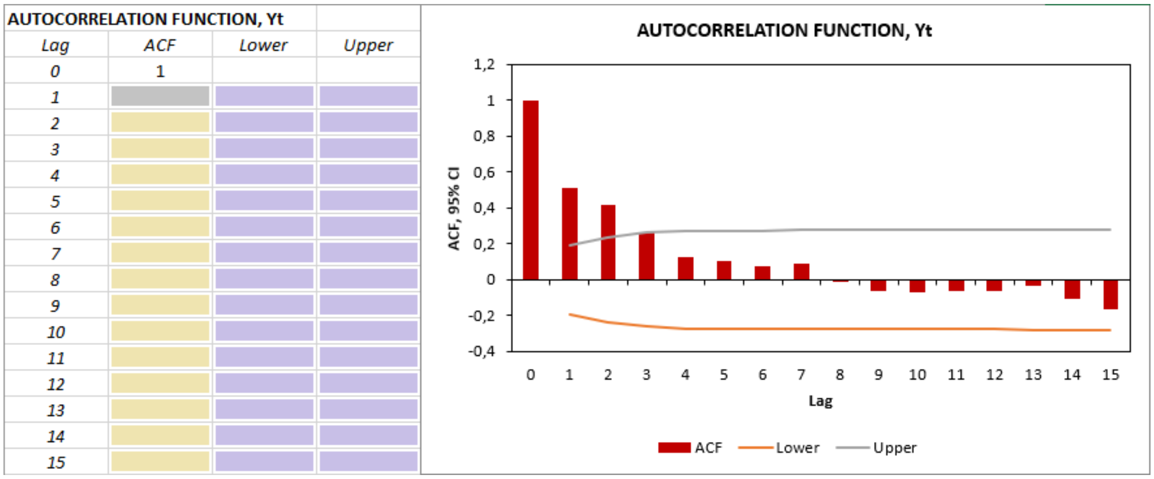
[v0.0-v0.1]

ra_AutoCorr_ACF(range_Yt, int_Lag)

Returns the AutoCorrelation Function at lag k, FAC(k), for a time series in range Yt.

$$FAC(k) = \rho_k = Cov(Y_t, Y_{t-k})/Var(Y_t)$$

- **range_Yt,** A column or range of a time series.
- **int_Lag,** Positive integer value (>=0).



AUTOCORRELATION FUNCTION, Yt

Lag	*ACF*	*Lower*	*Upper*
0	1		
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			

ra_AutoCorr_ACF_Test(range_Yt, int_MaxLag)

Returns the significance test for the FAC autocorrelation function, z-Stat and p-Value (probability) of a time series in the range Yt.

- **range_Yt,** A column or range of a time series.
- **int_MaxLag,** Positive integer value (>=0).

ra_AutoCorr_ACF_CI(range_Yt, int_MaxLag)

Returns the Confidence Interval (CI) of the Autocorrelation Function (ACF) of a time series in the range Yt.

- **range_Yt,** A column or range of a time series.
- **int_MaxLag,** Positive integer value (>=0).

ra_AutoCoVar_ACVF(range_Yt, int_Lag)

Returns the value of the AutoCoVariance Function (ACVF(k)*(Obs-1)) of a time series in the range Yt

- **range_Yt,** A column or range of a time series.
- **int_Lag,** Positive integer value (>=0).

ra_AutoCoVar_ACVF_Matrix(range_Yt, int_MaxLag)

Returns the AutoCoVariance Function (ACVF(k)) matrix of a time series in the range Yt.

- **range_Yt,** A column or range of a time series.
- **int_MaxLag,** Positive integer value (>=0).

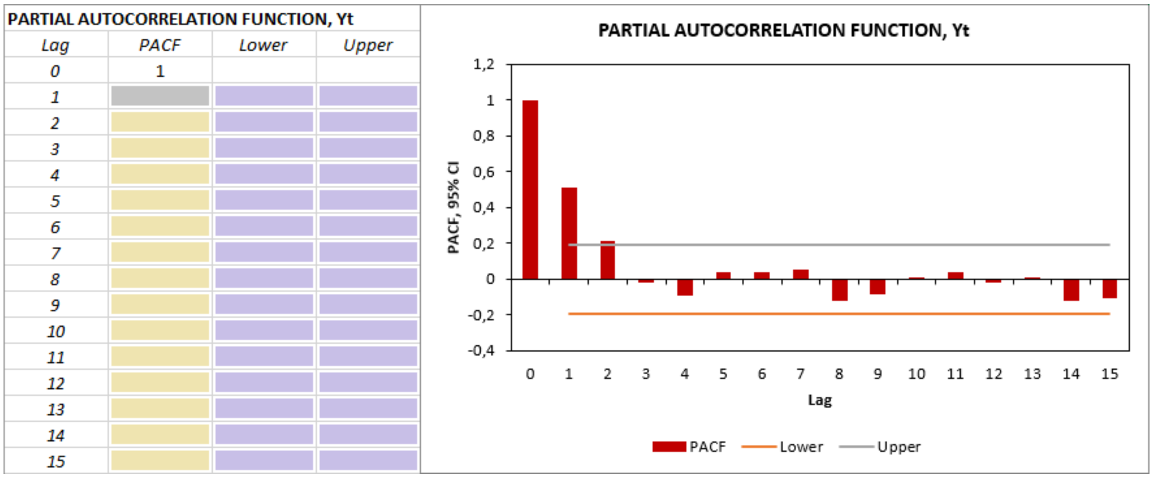
$$\begin{bmatrix} var(Yt) & \cdots & Cov(Yt, Yt-k) \\ \vdots & \ddots & \vdots \\ Cov(Yt, Yt-k) & \cdots & var(Yt) \end{bmatrix}$$

ra_Partial_AutoCorr_PACF(range_Yt, int_Lag)

Returns the Partial AutoCorrelation Function at lag k, ACPF(k), for the time series in range Yt.

$$FACP(k) = Corr(Y_t, Y_{t-k} | Y_{t-1}, Y_{t-2}, \dots Y_{t-k-1})$$

- **range_Yt,** A column or range of a time series.
- **int_Lag,** Positive integer value (>=0).



PARTIAL AUTOCORRELATION FUNCTION, Yt

Lag	PACF	Lower	Upper
0	1		
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			

ra_Partial_AutoCorr_PACF_CI(range_Yt, int_MaxLag)

Returns the Confidence Interval (CI) of the partial autocorrelation function (PACF) of a time series in the range Yt.

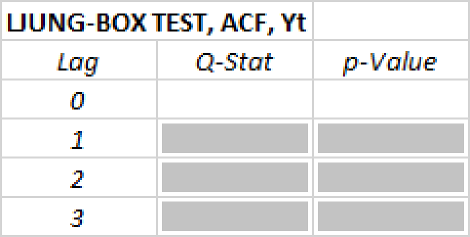
- **range_Yt,** A column or range of a time series.
- **int_MaxLag,** Positive integer value (>=0).

ra_LjungBox_Test(range_Yt, int_Lag, bool_ACF)

Returns the Q-Stat statistic and p-Value (probability) of the Ljung-Box test of a time series in the range Yt.

$$Ljung - Box(Q) = n(n+2) \sum_{j=1}^{m} \frac{\rho_j^2}{n-j} \sim X^2(m)$$

- **range_Yt,** A column or range of a time series.
- **int_Lag,** Positive integer (>=0).
- **bool_ACF,** If TRUE is for ACF, if FALSE is for PACF.



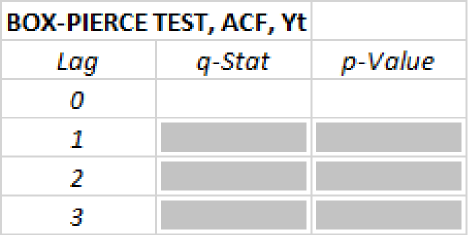
LJUNG-BOX TEST, ACF, Yt		
Lag	*Q-Stat*	*p-Value*
0		
1		
2		
3		

ra_BoxPierce_Test(range_Yt, int_Lag, bool_ACF)

Returns the Box-Pierce test statistics and p-Values (probability) for a time series in the range Yt.

$$Box - Pierce(q) = n \sum_{j=1}^{m} \rho_j^2 \sim X^2(m)$$

- **range_Yt,** A column or range of a time series.
- **int_Lag,** Positive integer (>=0).
- **bool_ACF,** If TRUE is for ACF, if FALSE is for PACF.



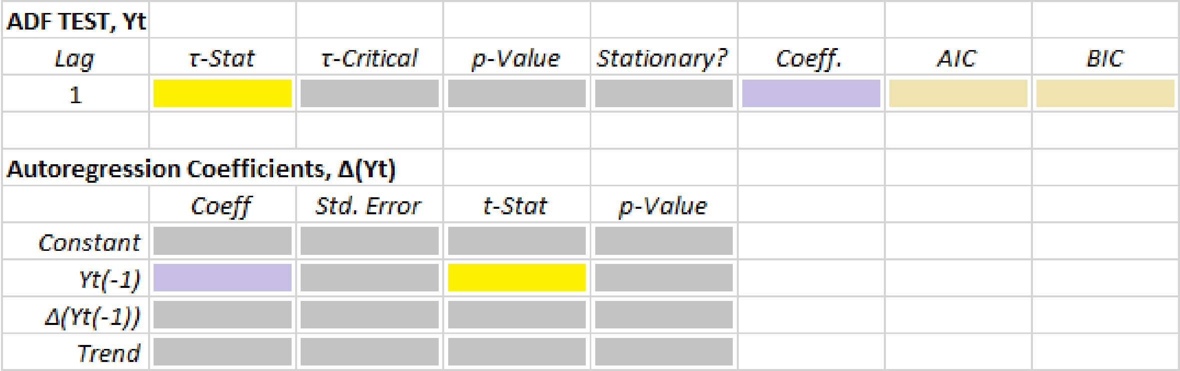
BOX-PIERCE TEST, ACF, Yt		
Lag	*q-Stat*	*p-Value*
0		
1		
2		
3		

ra_DickeyFuller_ADF_Test(range_Yt, bool_Ascend_Yt, int_Lag, int_Type, alpha)

Returns the τ-Stat, τ-Critical, p-Value (probability), and whether stationary + [AutoRegression Coefficient, AIC, and BIC] for ADF (Augmented Dickey-Fuller) test.

$$\Delta Y_t = \mu + \lambda Y_{t-1} + \sum_{i=1}^{p} \delta_i \Delta Y_{t-i} + \beta T$$

- **range_Yt,** A column or range of a time series.
- **bool_Ascend_Yt,** is the order of the time series. TRUE means ascending (↑) order with the most recent date on top. FALSE otherwise Descending (↓).
- **int_Lag,** positive integer value. 0 if Dickey-Fuller; if >=1 if Augmented Dickey-Fuller.
- **int_Type,** 0 if there is no constant or trend (nc); 1 if it is with constant and without trend (c); 2 if it is with constant and trend (ct).
- **Alpha,** significance level, alpha = [1%, 5%, 10%].



ADF TEST, Yt							
Lag	*τ-Stat*	*τ-Critical*	*p-Value*	*Stationary?*	*Coeff.*	*AIC*	*BIC*
1	🟨	▨	▨	▨	🟪	🟫	🟫
Autoregression Coefficients, Δ(Yt)							
	Coeff	*Std. Error*	*t-Stat*	*p-Value*			
Constant	▨	▨	▨	▨			
Yt(-1)	🟪	▨	🟨	▨			
Δ(Yt(-1))	▨	▨	▨	▨			
Trend	▨	▨	▨	▨			

ra_DickeyFuller_ADF_Critical(int_Obs, int_Type)

Returns the τ-Critical value for augmented Dickey-Fuller, alpha = [1%, 5%, 10%]. *MacKinnon (1996)

- **int_Obs,** Number of observations after adjustments or size of a time series.
- **int_Type,** 0 if there is no constant or trend (nc); 1 if it is with constant and without trend (c); 2 if it is with constant and trend (ct).

ra_DickeyFuller_ADF_pValue(t_Stat, int_Obs, int_Type)

Returns the approximate Dickey-Fuller p-Value for a time series of the Observations τ-Stat statistic.

- **t_Stat,** test statistic value or tau.
- **int_Obs,** Number of observations after adjustments or size of a time series.
- **int_Type,** 0 if there is no constant or trend (nc); 1 if it is with constant and without trend (c); 2 if it is with constant and trend (ct).

ra_DickeyFuller_ADF_Reg(range_Yt, bool_Ascend_Yt, int_Lag, int_Type)

Returns the AutoRegression Coefficient for the Augmented Dickey-Fuller (ADF) test.

- **range_Yt,** A column or range of a time series.
- **bool_Ascend_Yt,** is the order of the time series. TRUE means ascending (↑) order with the most recent date on top. FALSE otherwise Descending (↓).
- **int_Lag,** positive integer value. 0 if Dickey-Fuller; >=1 if Dickey-Fuller Augmented
- **int_Type,** 0 if there is no constant or trend (nc); 1 if it is with constant and without trend (c); 2 if it is with constant and trend (ct).

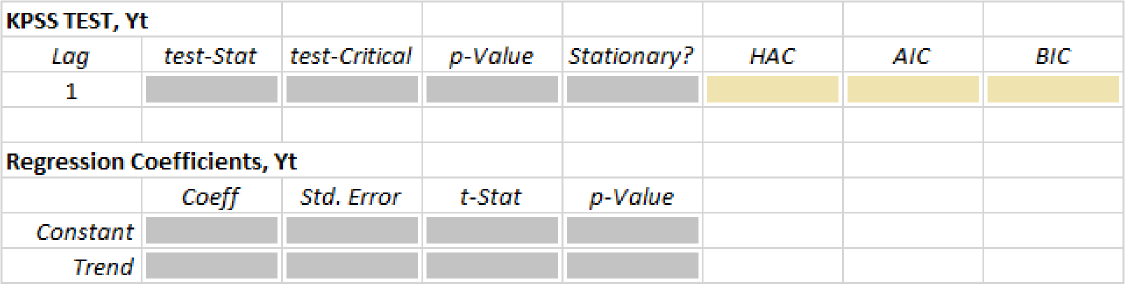
ra_KPSS_Test(range_Yt, bool_Ascend_Yt, int_Lag, int_Type, alpha)

Returns the test statistic test-Stat, the test-Critical, the p-Value (probability), and whether it is stationary + [variance corrected by HAC, AIC, and BIC] for the KPSS (Kwiatkowski-Phillips-Schmidt-Shin) test.

$$LM = \sum_{t=0}^{p} (Y_t - (\mu + \beta T))/(T^2 f_0)\,;\; f_0 = 1 - |\varepsilon_t|\ si\ \varepsilon_t \leq 1$$

- **range_Yt,** A column or range of a time series.

- **bool_Ascend_Yt,** is the order of the time series. TRUE means ascending (↑) order with the most recent date on top. FALSE otherwise Descending (↓).
- **int_Lag,** Positive integer value (>=0).
- **int_Type,** 1 if it is with constant and without trend (c); 2 if it is with constant and trend (ct).
- **Alpha,** significance level, alpha = [1%, 5%, 10%].



KPSS TEST, Yt							
Lag	*test-Stat*	*test-Critical*	*p-Value*	*Stationary?*	*HAC*	*AIC*	*BIC*
1							
Regression Coefficients, Yt							
	Coeff	*Std. Error*	*t-Stat*	*p-Value*			
Constant							
Trend							

ra_KPSS_Critical(int_Type)

Returns the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test-critical value, alpha = [1%, 5%, 10%]. *KPSS (1992, Table 1)

- **int_Type,** 1 if constant and no trend (c); 2 if constant and trend (ct).

ra_KPSS_pValue(t_Stat, int_Obs, t_Type)

Returns the approximate Dickey-Fuller p-Value (probability) for a time series of the test-Stat statistic of the Observations.

- **t_Stat,** test-Stat test statistic.
- **int_Obs,** Number of observations after adjustments, size of a time series.
- **t_Type,** 1 if it is with constant and without trend (c); 2 if it is with constant and trend (ct).

ra_KPSS_Reg(range_Yt, bool_Ascend_Yt, int_Lag, int_Type)

Returns the Regression Coefficients for the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test.

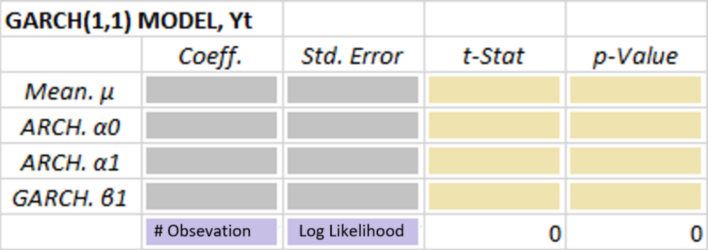
- **range_Yt,** A column or range of a time series.
- **bool_Ascend_Yt,** is the order of the time series. TRUE means ascending (↑) order with the most recent date on top. FALSE otherwise Descending (↓).
- **int_Lag,** Positive integer value (>=0).
- **int_Type,** 1 if it is with constant and without trend (c); 2 if it is constant and trend (ct).

ra_GARCH_Coeff(range_Yt, bool_Ascend_Yt, int_Alpha_p, int_Beta_q, bool_MeanCond, int_ErrDist, int_OptMethod)

Returns the coefficients, standard error, t-statistic, and p-Value (probability) for the Generalized AutoRegressive Conditional Heteroskedasticity GARCH(p,q) model.

$$Y_t = z\sigma_t$$
$$z \sim iidN(0,1)$$
$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i Y_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2$$

- **range_Yt,** A column or range of a time series.
- **bool_Ascend_Yt,** is the order of the time series. TRUE means ascending (↑) order with the most recent date on top. FALSE otherwise Descending (↓).
- **int_Alpha_p,** Number of ARCH Alpha(p) parameters (non-negative integer, >=0).
- **int_Beta_q,** Number of Beta(q) GARCH parameters (non-negative integer, >=0).
- **bool_MeanCond,** conditional mean. TRUE include in model. FALSE do not include.
- **int_ErrDist,** default normal (Gaussian) [8].
- **int_OptMethod,** default Newton-Raphson (NR).



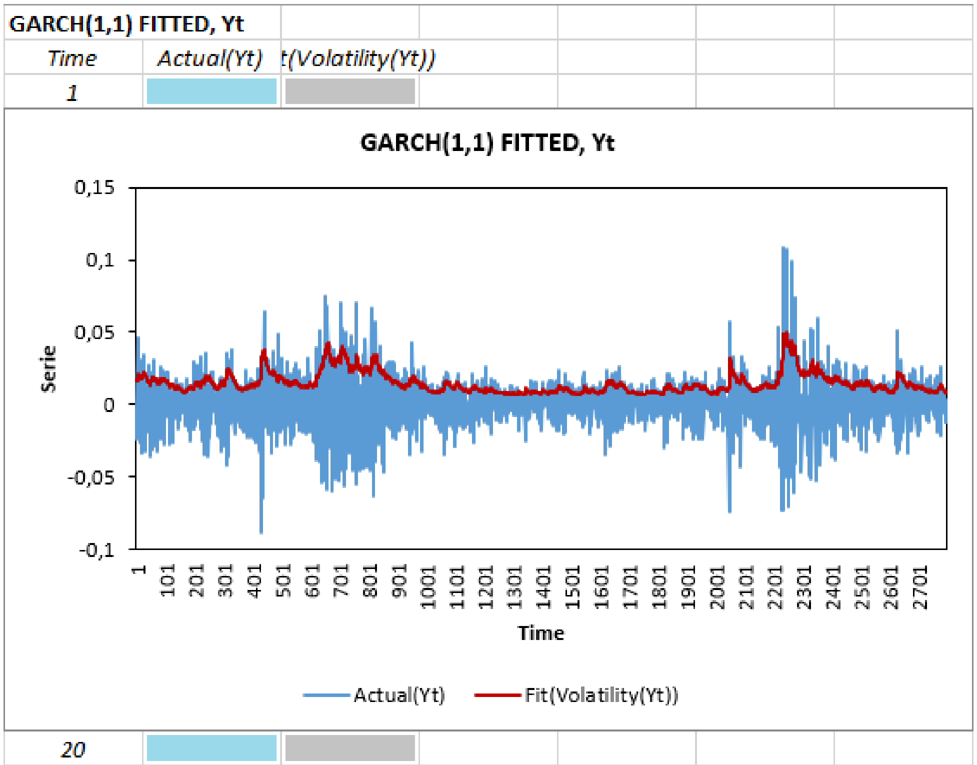
GARCH(1,1) MODEL, Yt				
	Coeff.	*Std. Error*	*t-Stat*	*p-Value*
Mean. μ				
ARCH. α0				
ARCH. α1				
GARCH. β1				
	# Obsevation	Log Likelihood	0	0

ra_GARCH_Fitted(range_Yt, bool_Ascend_Yt, range_Beta)

Returns a column with the adjusted Volatility of the Generalized AutoRegressive Conditional Heteroskedasticity model GARCH(p,q).

- **range_Yt,** A column or range of a time series.
- **bool_Ascend_Yt,** is the order of the time series. TRUE means ascending (↑) order with the most recent date on top. FALSE otherwise Descending (↓).
- **range_Alpha,** A column or range of ARCH (Alpha) coefficients.
- **range_Beta,** A column or range of GARCH (Beta) coefficients.

[8] For the GARCH model, the implementation of the t-Student probability and Generalized Error Distribution (GED) is pending.



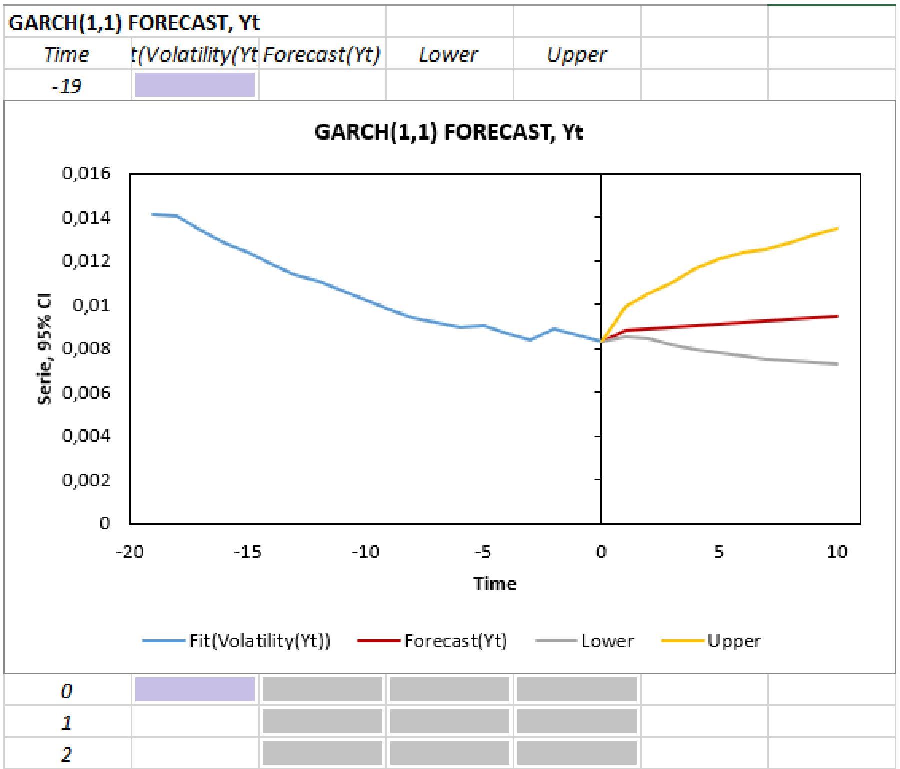
GARCH(1,1) FITTED, Yt

Time	Actual(Yt)	t(Volatility(Yt))
1		
20		

ra_GARCH_Forecast(range_Yt, bool_Ascend_Yt, range_Alpha, range_Beta, int_nForecast, int_Interval, int_IterSamples)

Returns a column or matrix of dynamic forecast volatility with confidence interval from the GARCH(p,q) model to a time series.

- **range_Yt,** A column or range of a time series.
- **bool_Ascend_Yt,** is the order of the time series. TRUE means ascending (↑) order with the most recent date on top. FALSE otherwise Descending (↓).
- **range_Alpha,** A column or range of ARCH (Alpha) coefficients.
- **range_Beta,** A column or range of GARCH (Beta) coefficients.
- **int_nForecast,** N-forecast number.
- **int_Interval,** confidence interval. 0 dynamic no interval, 1 Monte Carlo simulation for 68% interval, 2 for 95% interval, and 3 for 99.7% interval.
- **int_IterSamples,** Iteration number to estimate the confidence interval by the Monte Carlo method.



GARCH(1,1) FORECAST, Yt						
Time	*t(Volatility(Yt*	*Forecast(Yt)*	*Lower*	*Upper*		
-19						
0						
1						
2						

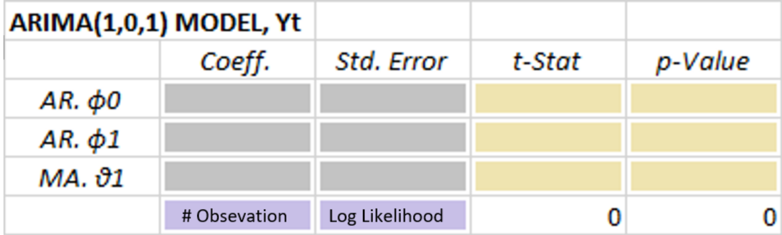
ra_ARIMA_Coeff(range_Yt, bool_Ascend_Yt, int_AR_p, int_Diff_d, int_MA_q, int_OptMethod)

Returns the coefficients, standard error, t-statistic, and p-Value (probability) of the AutoRegressive, Integrated, and Moving Average ARIMA(p,d,q) model.

- **range_Yt,** A column or range of a time series.
- **bool_Ascend_Yt,** is the order of the time series. TRUE means ascending (↑) order with the most recent date on top. FALSE otherwise Descending (↓).
- **int_AR_p,** Number of parameters AR(p) (non-negative integer, >=0).
- **int_Diff_d,** Order of integration(I) or number of Differentiations(d) (non-negative integer, >=0).
- **int_MA_q,** Number of MA(q) parameters (non-negative integer, >=0).
- **int_OptMethod,** AR(OLS) and MA(NR, Newton-Raphson)[9].

$$Y_t = \phi_0 + \sum_{i=1}^{p} \phi_i Y_{t-1} + \sum_{j=1}^{q} \theta_i \varepsilon_{t-j}$$

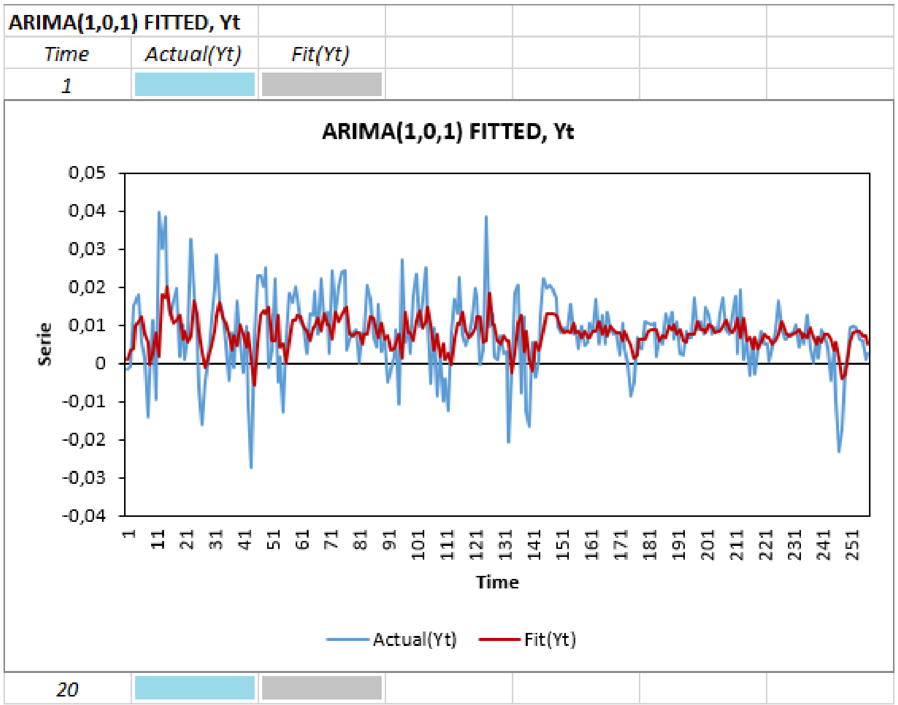
[9] For the AR model, it is estimated by Ordinary Least Squares (OLS) and for MA it is by Maximum Likelihood with iterative Newton-Raphson (NR). For the ARIMA, GARCH models or functions that require optimization, the implementation of optimization algorithms such as Levenberg-Marquardt, BHHH, BFGS or L-BFGS is planned.



ARIMA(1,0,1) MODEL, Yt				
	Coeff.	*Std. Error*	*t-Stat*	*p-Value*
AR. ϕ0				
AR. ϕ1				
MA. ϑ1				
	# Obsevation	Log Likelihood	0	0

ra_ARMA_Fitted(range_Yt, bool_Ascend_Yt, constant, range_AR, range_MA)
Returns a fitted column of the AutoRegressive and Moving Average ARIMA(p,q) model.

- **range_Yt,** A column or range of a time series.
- **bool_Ascend_Yt,** is the order of the time series. TRUE means ascending (↑) order with the most recent date on top. FALSE otherwise Descending (↓).
- **constant,** ARMA constant, AR, MA or average of MA.
- **range_AR,** Range of AutoRegressive (AR) Coefficients.
- **range_MA,** range of Moving Average (MA) coefficients.

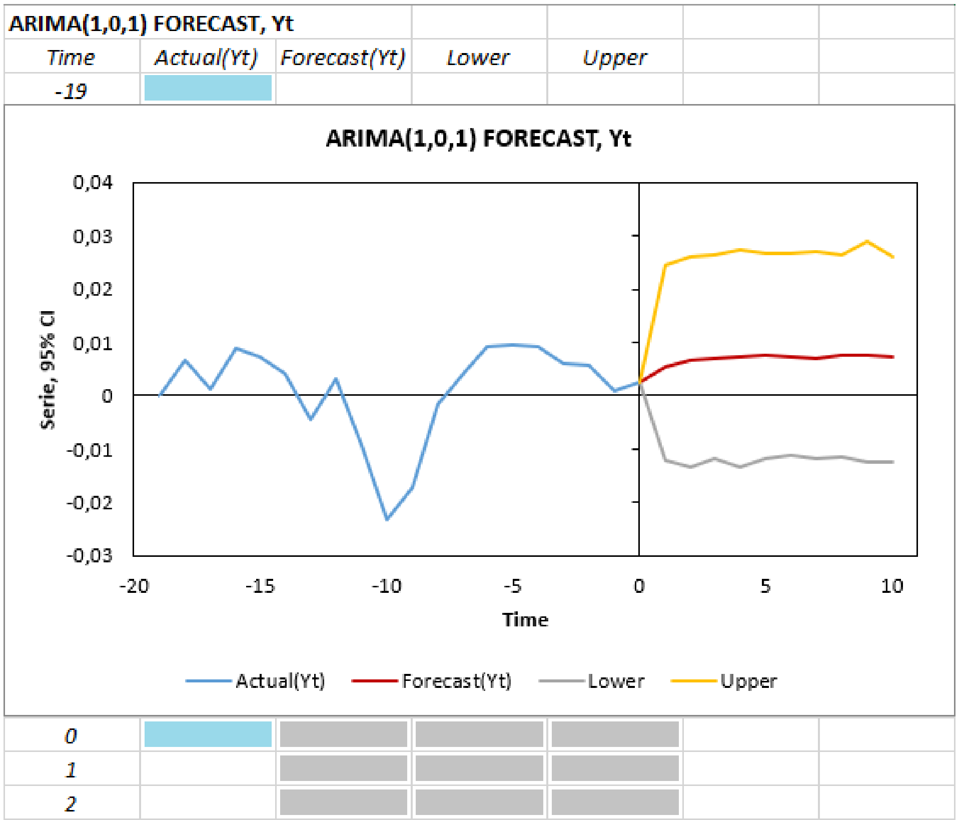


ARIMA(1,0,1) FITTED, Yt		
Time	*Actual(Yt)*	*Fit(Yt)*
1		
20		

ra_ARMA_Forecast(range_Yt, bool_Ascend_Yt, constant, range_AR, range_MA, int_nForecast, int_Interval, int_IterSamples)
Returns a dynamic forecast column or matrix with confidence interval from the ARMA(p,q) model of a time series.

- **range_Yt,** A column or range of a time series.
- **bool_Ascend_Yt,** is the order of the time series. TRUE means ascending (↑) order with the most recent date on top. FALSE otherwise Descending (↓).
- **constant,** Constant ARMA, AR, MA or MA Average.
- **range_AR,** Range AutoRegression (AR) coefficients.

- **range_MA,** Moving average (MA) range coefficients.
- **int_nForecast,** number of N-forecast steps.
- **int_Interval,** confidence interval. 0 dynamic no interval, 1 Monte Carlo simulation for 68% interval, 2 for 95% interval, and 3 for 99.7% interval.
- **int_IterSamples,** Iteration number to estimate the confidence interval by the Monte Carlo method.



ARIMA(1,0,1) FORECAST, Yt						
Time	*Actual(Yt)*	*Forecast(Yt)*	*Lower*	*Upper*		
-19						
0						
1						
2						

ra_InterpolateNum(x, x1, x2, y1, y2, h)

Returns the interpolation for y between x, x1, and x2 given y1 and y2.

$$\begin{Bmatrix} y_1 \\ y? \\ y_2 \end{Bmatrix} = \begin{Bmatrix} x_1 \\ x \\ x_2 \end{Bmatrix}$$

- **x,** number between x1 and x2.
- **x1,** numerical value.
- **x2,** numerical value.
- **y1,** numerical value.
- **y2,** numerical value.
- **int_Type,** 0 for linear interpolation (default); 1 for logarithmic interpolation; 2 for harmonic interpolation.

ra_Range_Flip(range_Xij, bool_Flip)

Returns a flipped range of cells vertically if the condition is true.

- **range_Xij,** range containing numeric values without spaces. Could be a multi-column Xij range.
- **bool_Flip,** if equal to TRUE, flips the range of cells in the vertical direction.

ra_Show_Lag(range_Yt, bool_Ascend_Yt, int_Lag)

Displays a lagging column of a time series.

- range_Yt, A column or range of a time series.
- **bool_Ascend_Yt,** is the order of the time series. TRUE means ascending (↑) order with the most recent date on top. FALSE otherwise Descending (↓).
- **int_Lag,** Positive integer value (>=0).

ra_Difference(range_Yt, bool_Ascend_Yt, int_Diff_d)

Returns a column difference operation on a time series.

- **range_Yt,** A column or range of a time series.
- **bool_Ascend_Yt,** is the order of the time series. TRUE means ascending (↑) order with the most recent date on top. FALSE otherwise Descending (↓).
- **int_Diff_d,** Order of integration(I) or number of differentiations(d) (non-negative integer, >=0)

ra_raXL_Stat_License()

Displays details of the current or used license.

ra_raXL_Stat_Version()

Lists supported and tested Excel versions of raXL Stat.

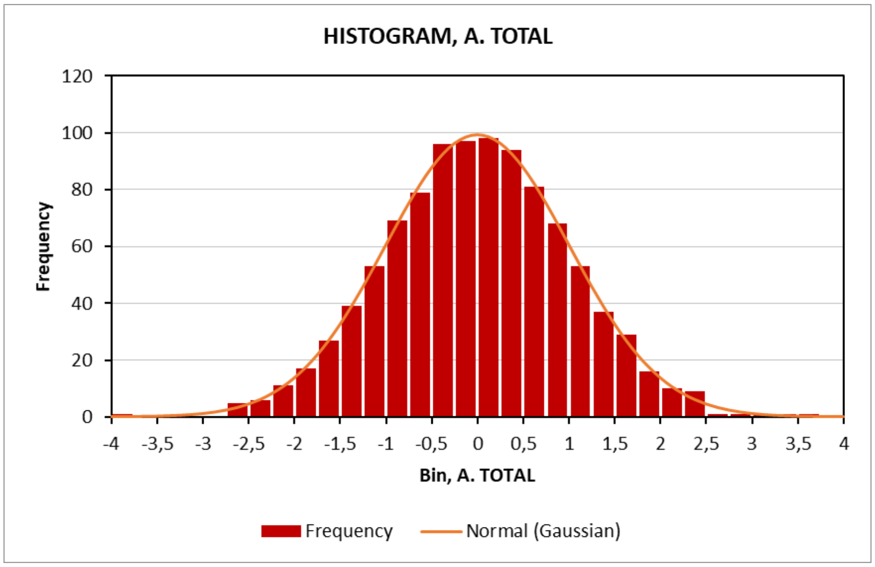
ra_raXL_xlDCPU()

Reserved for raXL_Stat.lic

[v0.2]

ra_Histogram_Auto(range_X, int_CurveSelect, bool_Freq)

Returns a frequency table with automatic, cumulative or normal curve bins, useful for plotting a histogram. [Obs. > 4]



- **range_X,** A range of cells in one or more columns that contains data.
- **int_CurveSelect,** Type of curve overlay. 0 if no curve overlay. 1 if 'cumulative probability' is displayed. 2 if 'cumulative unit' and 3 if 'Normal (Gaussian)' is displayed.
- **bool_Freq,** if TRUE uses 'Frequency' (Absolute) values. If FALSE uses 'Density' (Relative Frequency) values.

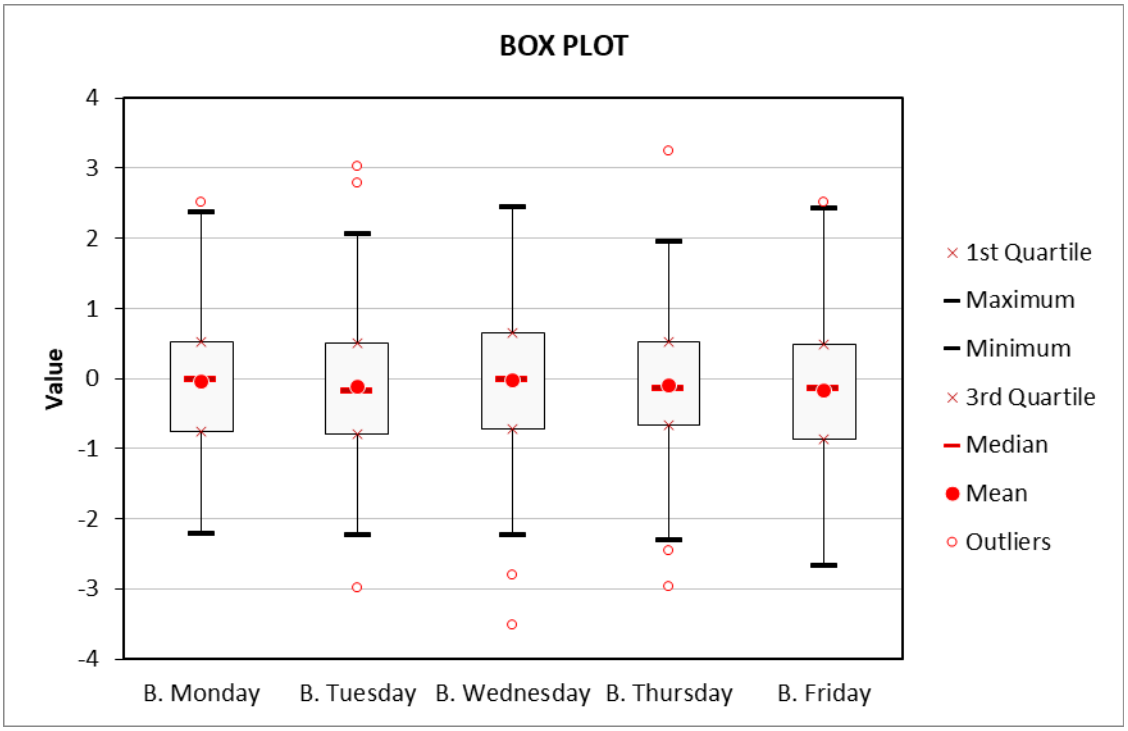
ra_Histogram_Bin(range_X, int_CurveSelect, bool_Freq, val_XStart, val_XStep, val_XStop)

Returns a frequency table with bins, cumulative or normal curve settings, useful for plotting a histogram.

- **range_X,** A range of cells in one or more columns that contains data.
- **int_CurveSelect,** Type of curve overlay. 0 if no curve overlay. 1 if 'cumulative probability' is displayed. 2 if 'cumulative unit' and 3 if 'Normal (Gaussian)' is displayed.
- **bool_Freq,** if TRUE uses 'Frequency' (Absolute) values. If FALSE uses 'Density' (Relative Frequency) values.
- **val_XStart,** New starting (minimum) value ranges on the horizontal axis.
- **val_XStep,** Largest unit of step between the start and end ranges.
- **val_XStop,** New end (maximum) value ranges on the horizontal axis.

ra_BoxPlot_Table(range_X, val_Outliers, val_IQR)

Returns a table of quartile values [1st quartile, maximum, minimum, 3rd quartile, median, mean, and outliers] useful for plotting a box-and-outlier plot.



- **range_X,** A range of cells in one or more columns that contains data.
- **val_Outliers,** If TRUE includes outliers. If FALSE does not include outliers.
- **val_IQR,** If TRUE, outliers are included. Value of IQR (interquartile range) multipliers, for example 1.5.

ra_JarqueBera_Test(range_X, bool_Pop, val_Alpha)

Return JB-Stat, p-Value (probability) and normality? for Jarque-Bera normality test.

$$Jarque - Bera = n\left(\frac{S^2}{6} + \frac{(K-3)^2}{24}\right) \sim \chi^2(2)$$

- **range_X,** A range of cells in one or more columns that contains data.
- **bool_Pop,** if TRUE, uses the skewness and kurtosis values from 'Population'. If FALSE, uses the skewness and kurtosis values from 'Sample'.
- **val_Alpha,** Significance level, alpha = [1%, 5%, 10%].

ra_JarqueBera_Stat(range_X, bool_Pop)

Returns the Jarque-Bera statistic (JB-Stat), for the Jarque-Bera normality test.

- **range_X,** A range of cells in one or more columns that contains data.
- **bool_Pop,** if TRUE, uses the skewness and kurtosis values from 'Population'. If FALSE, uses the skewness and kurtosis values from 'Sample'.

ra_JarqueBera_pValue(val_JB_Stat)

Returns the p-value (probability) for the Jarque-Bera normality test, Chi2(JB-Stat,2)

- **val_JB_Stat,** Jarque-Bera statistical value.

ra_Skew_S(range_X)

Returns the skewness value of the distribution based on the sample.

- **range_X,** A range of cells in one or more columns that contains data.

ra_Kurt_S(range_X)

Returns the kurtosis value of the distribution based on the sample.

- **range_X,** A range of cells in one or more columns that contains data.

ra_Skew_P(range_X)

Returns the skewness value of the distribution based on the population.

- **range_X,** A range of cells in one or more columns that contains data.

ra_Kurt_P(range_X, bool_Subtract3)

Returns the kurtosis value of the population-based distribution.

- **range_X,** A range of cells in one or more columns that contains data.
- bool_Subtract3, if TRUE subtracts to generate a kurtosis for a zero normal distribution. If FALSE do not subtract.

ra_AndersonDarling_Test(range_X, int_Dist, val_Alpha)

Returns the AD-Stat, p-value (probability), and distribution hypothesis for the one-sample Anderson-Darling test.

$$AD^2 = -N - S/N$$

$$S = \sum_{i=1}^{N} (2i-1)(log(X_i) + log(1 - X_{n-i+1}))$$

- **range_X,** A range of cells in one or more columns that contains data.
- **int_Dist,** Type of distribution. If 0, 'Generic' distribution, if 1, 'Normal', if 2, 'Unmodified Normal', if 3, 'LogNormal'.
- **val_Alpha,** Significance level, alpha = [1%, 5%, 10%].

ra_AndersonDarling_Stat(range_X, int_Dist)

Returns the Anderson-Darling test statistic (AD-Stat) for the theoretical test on range data.

- **range_X,** A range of cells in one or more columns that contains data.
- **int_Dist,** Type of distribution. If 0, 'Generic' distribution, if 1, 'Normal', if 2, 'Unmodified Normal', if 3, 'LogNormal'.

ra_AndersonDarling_pValue(val_AD_Stat, int_Dist)

Returns the estimated p-value (probability) for the one-sample Anderson-Darling test.

- **val_AD_Stat,** Anderson-Darling statistical value.

- **int_Dist,** Type of distribution. If 0, 'Generic' distribution, if 1, 'Normal', if 2, 'Unmodified Normal', if 3, 'LogNormal'.

ra_ShapiroWilk_Test(range_X, int_Dist, val_Alpha)

Returns the SW-Stat statistic, p-value (probability), and normality for the Shapiro-Wilk normality test. Uses Royston's algorithm, Obs[2;5000]

$$SW = \frac{\sum_{i=1}^{n}(a_i x_i)^2}{\sum_{i=1}^{n}(x_i - mu)^2}$$

- **range_X,** A range of cells in one or more columns that contains data.
- **val_Alpha,** Significance level, alpha = [1%, 5%, 10%].

ra_ShapiroWilk_Stat(range_X, int_Dist, val_Alpha)

Returns the Shapiro-Wilk statistic (SW-Stat) of the Shapiro-Wilk normality test. Uses Royston's algorithm, Obs[2;5000])

- **range_X,** A range of cells in one or more columns that contains data.

ra_ShapiroWilk_pValue(val_SW_Stat, int_Obs)

Returns the p-value (probability) for the Shapiro-Wilk normality test.

- **val_SW_Stat,** Shapiro-Wilk statistical value.
- **int_Obs,** Number of observations in the data range or time series.

ra_Descriptive_Stats(range_X)

Returns a table of descriptive statistics for a sample of a data range or a time series.

- **range_X,** A range of cells in one or more columns that contains data.

ra_CoVar_Matrix(range_X, bool_Pop)

Returns the population or sample covariance (Cov) matrix of multiple data in a range or time series.

- **range_X,** A range of cells in one or more columns that contains data.
- **bool_Pop,** if TRUE, uses the skewness and kurtosis values from 'Population'. If FALSE, uses the skewness and kurtosis values from 'Sample'.

ra_Correl_Coeff_Matrix(range_X, int_LowTriang, val_Alpha)

Returns the matrix of Pearson correlation coefficients (R, R^2, p-value and no correlation (R=0)?) of multiple data in a range or time series.

- **range_X,** A range of cells in one or more columns that contains data.
- **int_LowTriang,** Lower triangular matrix. If 0, R correlation coefficients. If 1, R-squared coefficients. If 2, p-values for R based on t-Stat. If 3, no correlation (R=0)?
- **val_Alpha,** Significance level, alpha = [1%, 5%, 10%].

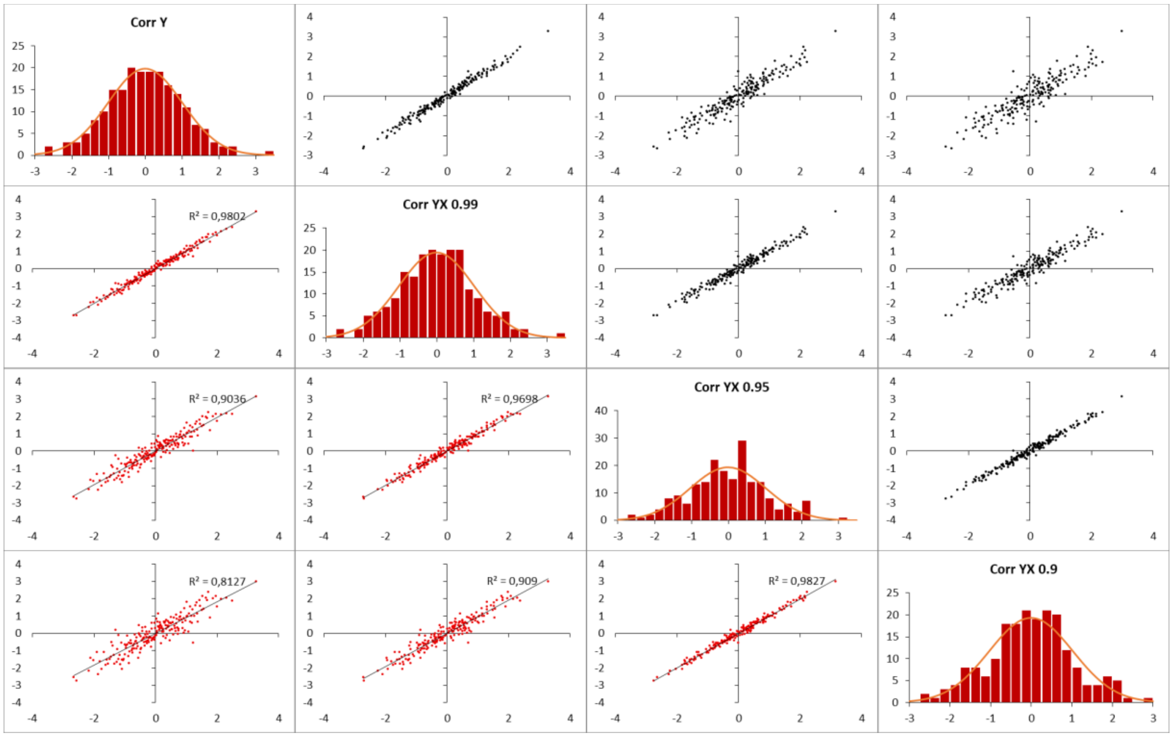
ra_MissingDataINFO(range_X)

Returns the count of missing data (blanks) and information for incorrect type values [Total Range, Numeric/Date, Empty/Missing, String, Boolean, ExcelError, Unheard, and Non-Numeric].

- **range_X,** A range of cells in one or more columns that contain data in which missing (blank) values or values of incorrect type might be present.

Scatter plot matrix

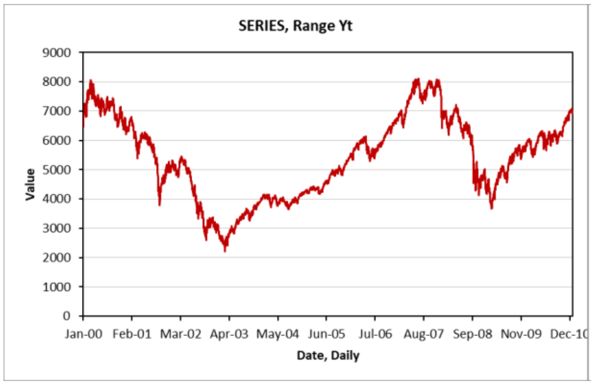
Displays the XY scatter matrix graph and histogram including normal or cumulative curve. The scatter matrix graph or diagram has no function except to make the histogram.

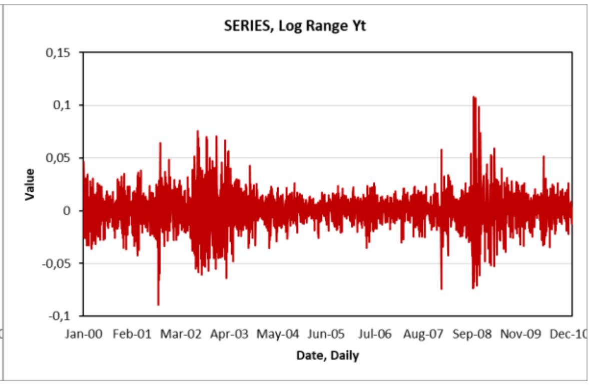


Multiple series plots

Displays multi-series charts of Line/Symbol by date, Column, XY line, Bar, Area and variants. For multi-series chart, it does not have an except function.

To run these charts, especially in time series, you need to have the first column contain time data (12/30/2024) and then select to display the summarized frequency based on the data in days or years on the x-axis.





[v0.3]

ra_Portfolio_Vector_Means (range_Returns)

Returns a vector containing the average returns of a portfolio.

- **range_Returns,** A single- or multi-column range of cells that contains data or returns.

ra_Portfolio_Vector_StdDev(matrix_Covariance)

Returns a vector of standard deviation of a portfolio.

- **matrix_Covariance,** Covariance matrix of the returns.

ra_Portfolio_Risk(matrix_Covariance, vector_Weights)

Returns the expected risk E[σ], volatility or Sigma (standard deviation) of a portfolio.

$$Variance = (w_1 \dots w_n)\begin{pmatrix} Cov_{11} & \cdots & Cov_{1n} \\ \vdots & \ddots & \vdots \\ Cov_{n1} & \cdots & Cov_{nn} \end{pmatrix}\begin{pmatrix} w_1 \\ \dots \\ w_n \end{pmatrix}$$

$$Sigma = \sqrt{Variance} = Risk = E[\sigma]$$

- **matrix_Covariance,** Covariance matrix of the returns.
- **vector_Weights,** Vector of weights of a portfolio.

ra_Portfolio_Return(vector_Mean_Returns, vector_Weights)

Returns the expected return E[r] of a portfolio.

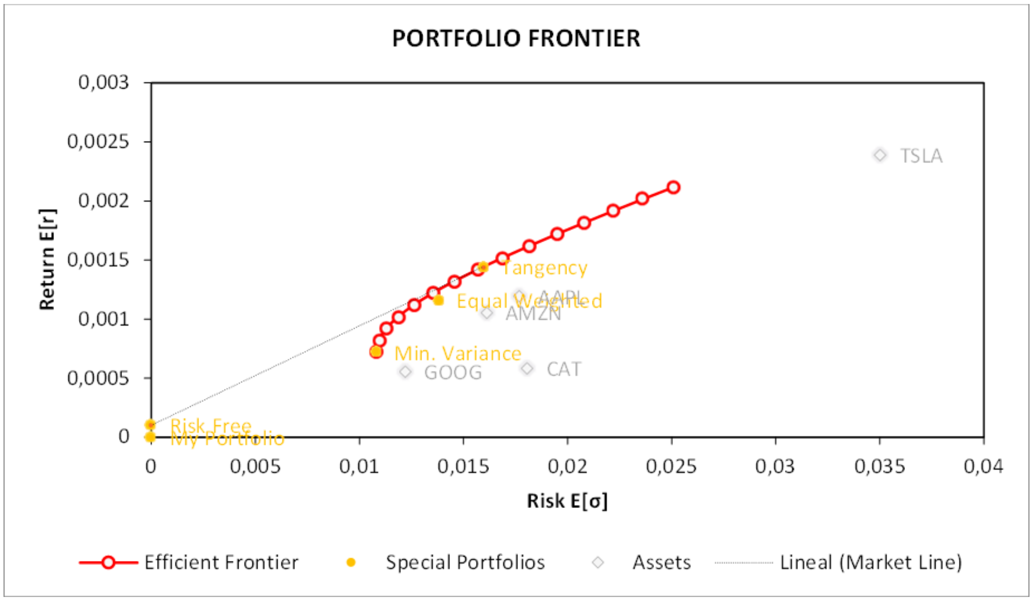
$$Return = (u_1 \dots u_n)\begin{pmatrix} w_1 \\ \dots \\ w_n \end{pmatrix} = E[r]$$

- **vector_Mean_Returns,** Average of the asset returns.
- **vector_Weights,** Vector of weights of a portfolio.

ra_Portfolio_Weights_Optimal(vector_Mean_Returns, matrix_Covariance, val_Expected_Return)

Return the weights of investments in a portfolio on the efficient frontier, using the Markowitz (1959) and Merton (1972) framework.

$$\begin{gathered} Maximize\ w^T u - \lambda w^T \Sigma w \\ subject\ to\ 1^T w = 1 \end{gathered}$$



- **vector_Mean_Returns,** Average of the asset returns.
- **matrix_Covariance,** Covariance matrix of the returns.
- **val_Expected_Return,** Required return or expected return E[r].

ra_Portfolio_Weights_Tangency(vector_Mean_Returns, matrix_Covariance, val_Risk_free)

Returns the tangency of the weights of investments in a portfolio on the efficient frontier. Using the Markowitz (1959) and Merton (1972) framework.

- **vector_Mean_Returns,** Average of the asset returns.
- **matrix_Covariance,** Covariance matrix of the returns.
- **val_Risk_free,** Risk-free rate.

ra_Portfolio_Market_Line_Effic(vector_Mean_Returns, matrix_Covariance, val_Risk_free, int_Frontier_Points)

Returns the capital market line (CML) on an efficient frontier portfolio. Using the Markowitz (1959) and Merton (1972) framework.

- **vector_Mean_Returns,** Average of the asset returns.
- **matrix_Covariance,** Covariance matrix of the returns.
- **val_Risk_free,** Risk-free rate.
- **int_Frontier_Points,** Number of points to display on the efficient frontier.

ra_Portfolio_Market_Line_Opti(vector_Mean_Returns, matrix_Covariance, val_Risk_free, val_Expected_Return)

Returns the optimal capital market line (CML) on the efficient frontier, using the Markowitz (1959) and Merton (1972) framework.

- **vector_Mean_Returns,** Average of the asset returns.
- **matrix_Covariance,** Covariance matrix of the returns.
- **val_Risk_free,** Risk-free rate.
- **val_Expected_Return,** Required return or expected return E[r].

ra_Portfolio_Weights_MinVar(vector_Mean_Returns, matrix_Covariance)

Returns the weights of minimum variance [MinVar] investments in a portfolio on the efficient frontier. Using the Markowitz (1959) and Merton (1972) framework.

- **vector_Mean_Returns,** Average of the asset returns.
- **matrix_Covariance,** Covariance matrix of the returns.

ra_Portfolio_Frontier_Efficient(vector_Mean_Returns, matrix_Covariance, int_Frontier_Points)

Returns the expected return E[r] and risk E[Sigma] on the efficient portfolio frontier (EPF). Using the Markowitz (1959) and Merton (1972) framework.

- **vector_Mean_Returns,** Average of the asset returns.
- **matrix_Covariance,** Covariance matrix of the returns.
- **int_Frontier_Points,** Number of points to display on the efficient frontier.

ra_Portfolio_Frontier_Optimal(vector_Mean_Returns, matrix_Covariance, int_Frontier_Points)

Returns the optimal expected return E[r] and risk E[Sigma] of a portfolio using the Markowitz (1959) and Merton (1972) framework.

- **vector_Mean_Returns,** Average of the asset returns.
- **matrix_Covariance,** Covariance matrix of the returns.
- **int_Frontier_Points,** Number of points to display on the efficient frontier.

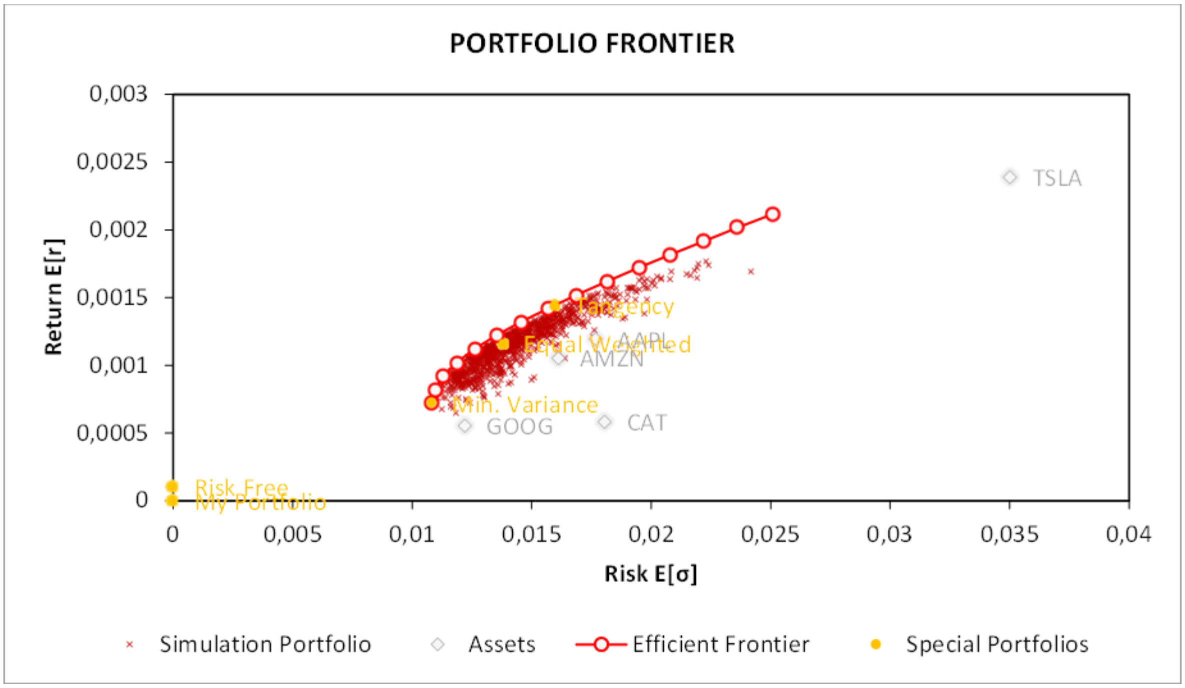
ra_Portfolio_Risk_Optimal(vector_Mean_Returns, matrix_Covariance, val_Expected_Return)

Returns the optimal expected risk E[σ] or Sigma (standard deviation) of the portfolio on the efficient frontier using the Markowitz (1959) and Merton (1972) framework.

- **vector_Mean_Returns,** Average of the asset returns.
- **matrix_Covariance,** Covariance matrix of the returns.
- **val_Expected_Return,** Required return or expected return E[r].

ra_Portfolio_Simulation(vector_Mean_Returns, matrix_Covariance, int_IterSamples, int_Seed)

Returns the [Risk, Return] points of a simulation of a portfolio.



- **vector_Mean_Returns,** Average of the asset returns.
- **matrix_Covariance,** Covariance matrix of the returns.
- **int_IterSamples,** Number of iterations to simulate portfolio points using the Monte Carlo method.
- **int_Seed,** Random seed for the Monte Carlo method. Example 1234.

ra_Portfolio_Simul_Frontier(vector_Mean_Returns, matrix_Covariance, int_IterSamples, int_Seed), int_Level_Zoom)

Returns the efficient points [Risk, Return] of a simulation of a portfolio. [Warning: Experimental, suboptimal weights!]

- **vector_Mean_Returns,** Average of the asset returns.
- **matrix_Covariance,** Covariance matrix of the returns.
- **int_IterSamples,** Number of iterations to simulate portfolio points using the Monte Carlo method.
- **int_Seed,** Random seed for the Monte Carlo method. Example 1234.
- **int_Level_Zoom,** Zoom level of points on the border. example 1, 2..

ra_Means_Column(range_X)

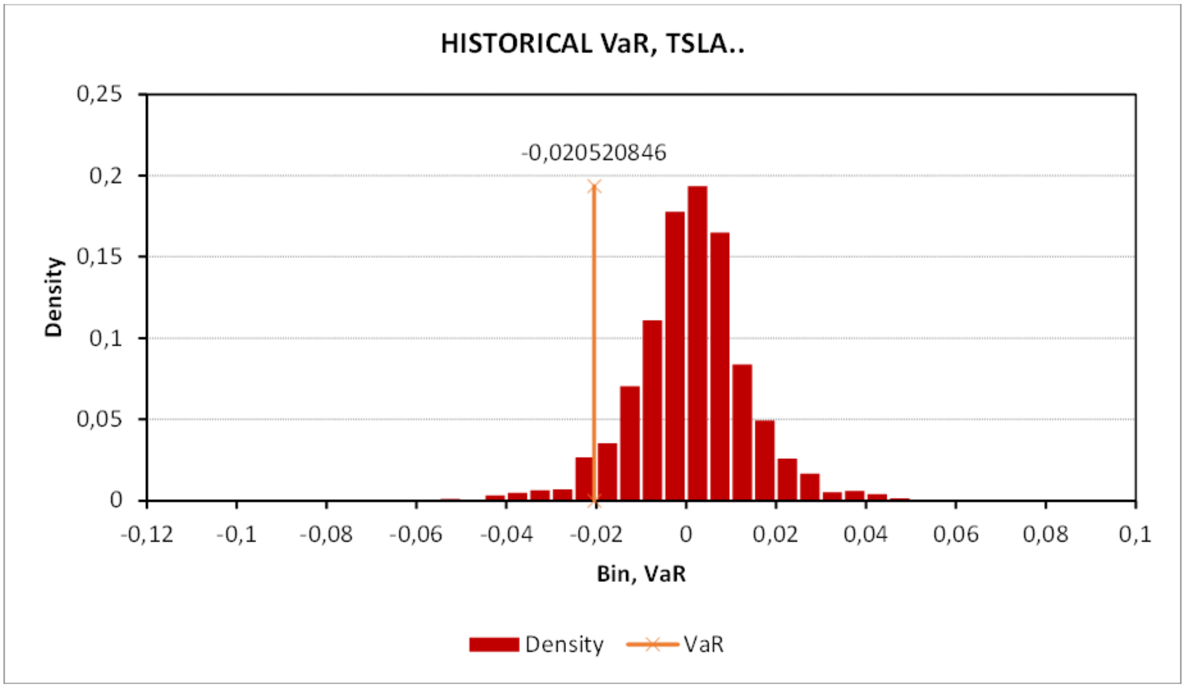
Returns a column of averages.

- **range_X,** A multi-column range of cells containing data or time series.

ra_VaR_Historical(range_Returns, val_Alpha, bool_Exc)

Returns the historical value at risk (H VaR) of a portfolio.

$$H\,VaR(\alpha) = percentile(r,\,1-\alpha)$$



- **range_Returns,** A single or multi-column range of cells containing data, returns, or time series.
- **val_Alpha,** Significance level, alpha = [1%, 5%, 10%].
- **bool_Exc,** for percentile. if TRUE 'Excludes' the first and last values 0 and 1. if FALSE 'Includes' them.

ra_VaR_Parametric(range_Returns, val_Alpha, bool_Pop)

Returns parametric values of the value at risk (P VaR) of a portfolio. Assumes normality.

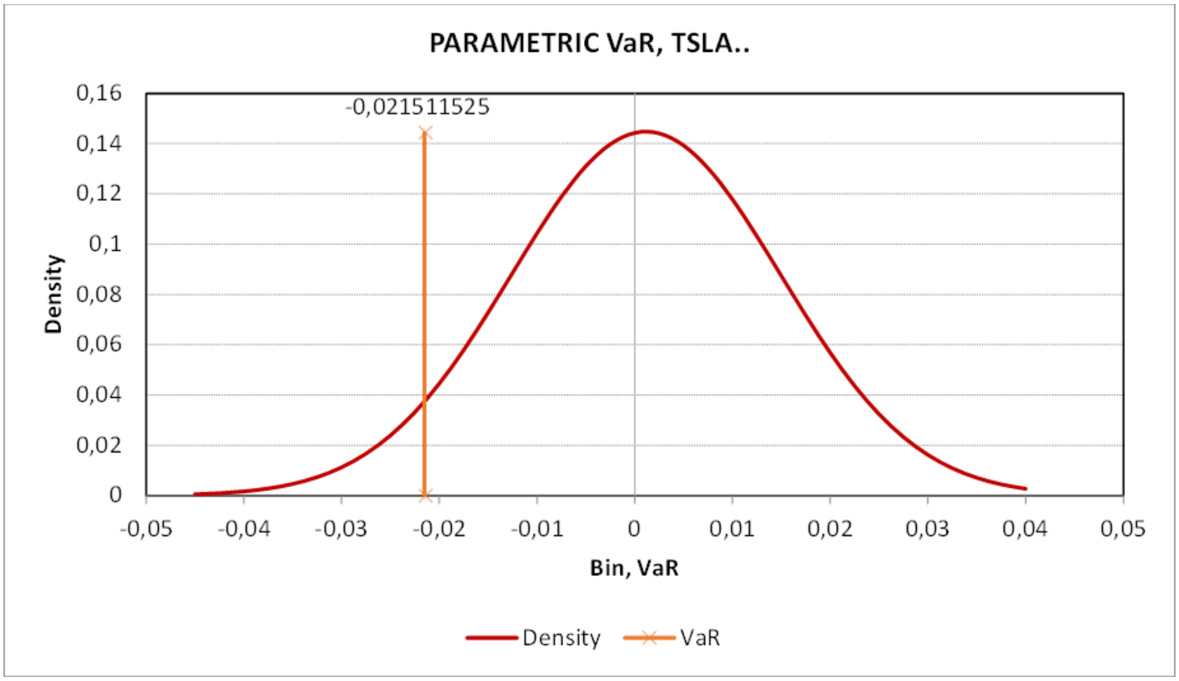
$$P\ VaR(\alpha) = (\mu + z(1 - \alpha) * \sigma)$$

- **range_Returns,** A single or multi-column range of cells containing data, returns, or time series.
- **val_Alpha,** Significance level, alpha = [1%, 5%, 10%].
- **bool_Pop,** for variance (Sigma^2). if TRUE for 'Population' variance values. if FALSE for 'Sample' variance values.

ra_VaR_VarCovar(vector_Mean_Returns, matrix_Covariance, vector_Weights, val_Alpha)

Returns the variance-covariance of the value at risk (P VaR) of a portfolio. Assumes normality.

$$VCV\ VaR(\alpha) = (\mu + z(1 - \alpha) * \sigma)$$



- **vector_Mean_Returns,** Average of the asset returns.
- **matrix_Covariance,** Covariance matrix of the returns.
- **vector_Weights,** Vector of weights of a portfolio.
- **val_Alpha,** Significance level, alpha = [1%, 5%, 10%].

ra_CVaR_Historical(range_Returns, val_Alpha, bool_Exc)

Returns the Historical Conditional Value at Risk (H CVaR) or Expected Shortfall of a portfolio.

- **range_Returns,** A single or multi-column range of cells containing data, returns, or time series.
- **val_Alpha,** Significance level, alpha = [1%, 5%, 10%].
- **bool_Exc,** for percentile. if TRUE 'Excludes' the first and last values 0 and 1. if FALSE 'Includes' them.

ra_CVaR_Parametric(range_Returns, val_Alpha, bool_Pop)

Returns conditional parametric values of the Value at Risk (P CVaR) or Expected Shortfall of a portfolio. Assumes normality.

- **range_Returns,** A single or multi-column range of cells containing data, returns, or time series.
- **val_Alpha,** Significance level, alpha = [1%, 5%, 10%].
- **bool_Pop,** for variance (Sigma^2). if TRUE for 'Population' variance values. if FALSE for 'Sample' variance values.

ra_CVaR_VarCovar(vector_Mean_Returns, matrix_Covariance, vector_Weights, val_Alpha)

Returns the conditional variance-covariance of the Value at Risk (P CVaR) or Expected Shortfall of a portfolio. Assumes normality.

- **vector_Mean_Returns,** Average of the asset returns.
- **matrix_Covariance,** Covariance matrix of the returns.
- **vector_Weights,** Vector of weights of a portfolio.
- **val_Alpha,** Significance level, alpha = [1%, 5%, 10%].

ra_Curve_Distribution(range_X, bool_Pop, int_Distr)

Returns a density or frequency table with automatic ranges or bins for plotting a curve.

- **range_X,** A multi-column range of cells containing data or time series.
- **bool_Pop,** for variance (Sigma^2). if TRUE for 'Population' variance values. if FALSE for 'Sample' variance values.
- **int_Distr,** Normal curve (Gaussian) Default.

ra_VaR_Backtesting_z(range_Returns, val_VaR, val_Alpha)

Returns a backtest vector of z-Stat, p-Value (Probability) and Adequate? for the VaR coefficient. H0: Is it accurate or adequate.

$$z = \frac{\hat{\alpha} - \alpha}{\sqrt{\frac{\alpha(1-\alpha)}{n}}} \sim N(0,1)$$

- **range_Returns,** A single or multi-column range of cells containing data, returns, or time series.
- **val_VaR,** Value at Risk (VaR) ratio.
- **val_Alpha,** Significance level, alpha = [1%, 5%, 10%].

ra_VaR_Backtesting_Kupiec(range_Returns, val_VaR, val_Alpha)

Returns a backtest vector of LR-Stat, p-Value (Probability) and Adequate? for the VaR coefficient. Kupiec (1995). H0: Is it accurate or adequate.

$$LR = 2\ln\left(\left(\frac{1-\hat{\alpha}}{1-\alpha}\right)^{T-I(\alpha)}\left(\frac{\hat{\alpha}}{\alpha}\right)^{I(\alpha)}\right) \sim \chi^2(1)$$

- **range_Returns,** A single or multi-column range of cells containing data, returns, or time series.
- **val_VaR,** Value at Risk (VaR) ratio.
- **val_Alpha,** Significance level, alpha = [1%, 5%, 10%].

5. Troubleshooting

Sometimes, there may be a problem with the activation or operation of the plugin. For example, the plugin may not be configured correctly or it may fail to update.

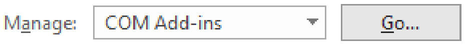
If you have a problem that is not described below or would like to discuss any aspect of the product, you can contact us at rubenfapaza@gmail.com.

However, the first step to troubleshooting is to make sure you have the latest version of the plugin installed. We regularly release fixes and new features at https://ruben-apaza.blogspot.com/p/raxl-stat.html, so it's important to keep your plugin or software up to date. Below are some common issues you may encounter:

5.1. The plugin menu tab or ribbon is not visible

After configuring the add-in, the “raXL Stat” tab should be visible within Excel (Figure 6). If it is not visible, the add-in may not have been activated correctly. To fix this:

- You can try to follow the steps described in 2.4. Download and use the plugin.
- If the above step does not work, click on the **File tab** in Excel:
- Then click on **Options** (located in the lower left corner):
- On the Options screen, select the **Add -ins** section.

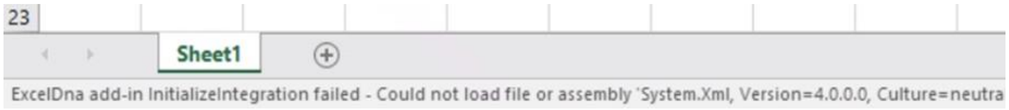


- Find the **View and Manage option** near the bottom, select "Excel COM Add-ins" and press "Go...":

You should see raXL Stat Add-in checked, make sure it is selected.

5.2. The add-in menu tab or ribbon is not visible (error message in Excel's bottom status bar)

When starting Excel after configuring the add-in, the tab is not visible. When checking the status bar at the bottom of Excel, the following error message appears:



ExcelDna add-in InitializeIntegration failed - Could not load file or assembly 'System.Xml, Version=4.0.0.0, Culture=neutra

(Error initializing ExcelDNA add-in integration: Could not load file or assembly 'System.Xml, Version=4.0.0.0, Culture=neutral, PublickKeyToken=....')

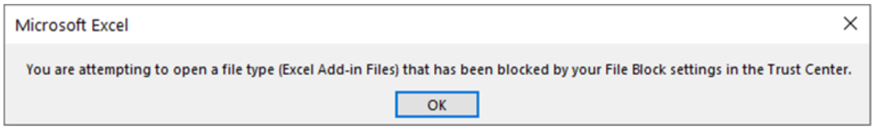
This error indicates that some of the components required for certain Microsoft applications to run on your PC, such as raXL Stat, are missing.

In this case, you need to download and install the Microsoft .NET Framework components from the Microsoft website by selecting the "Download .NET Framework 4.5.2" or later option described in 2.2. System Requirements

Once you have installed it we suggest you restart your PC and try to launch raXL Stat.

5.3. Problems starting Excel: You are trying to open a file type (Excel add-in file) that has been blocked by the File Block settings in the Trust Center.

When you start Excel after installing the add-in, you may see the following message:



You are attempting to open a file type (Excel Add-in Files) that has been blocked by your File Block settings in the Trust Center.

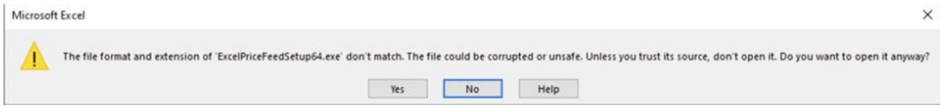
To resolve this, open the Excel Options screen by clicking the **File tab,** then **Options** (as shown at the top of this page) and navigate to "Trust Center."

Press the "Trust Center Settings..." button, then select "File Blocking Settings" on the left side.

Make sure the "Excel Add-in Files" checkbox is NOT checked.

5.4. Problems starting Excel: The file format and extension of the "add-in" do not match.

When you start Excel, after configuring the add-in, you may see the following message:



The file format and extension of 'ExcelPriceFeedSetup64.exe' don't match. The file could be corrupted or unsafe. Unless you trust its source, don't open it. Do you want to open it anyway?

This usually indicates that you have installed the wrong version of the add-in. For example, you may have inadvertently installed the 64-bit version of the add-in, while you are using the 32-bit version of Excel.

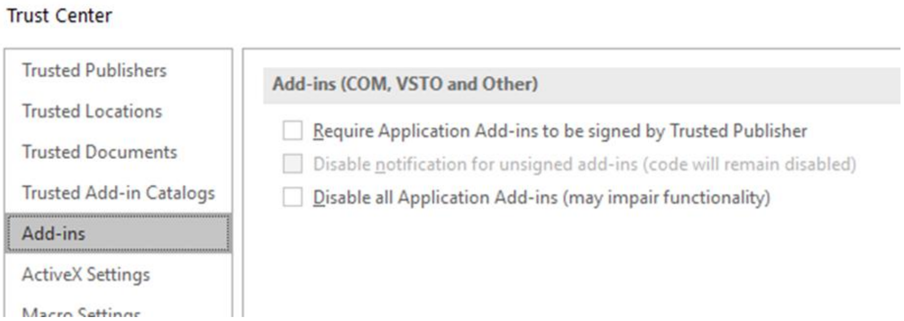
In this case, the first step is to open the correct add-in, described in 2.3. Excel version

5.5. The plugin formulas do not work and there are problems when starting

If Excel formulas aren't working and just return "#N/A" or you can't open them, the most likely cause is that Excel doesn't trust the add-in.

To resolve this, open the Excel Options screen by clicking the **File tab,** then **Options** (as shown at the top of this page) and navigate to "Trust Center."

Press the "Trust Center Settings..." button, then select "Add-ons" on the left side:



Add-ins (COM, VSTO and Other)

☐ Require Application Add-ins to be signed by Trusted Publisher

☐ Disable notification for unsigned add-ins (code will remain disabled)

☐ Disable all Application Add-ins (may impair functionality)

Make sure none of the three options are selected. If you select any of them, you may not be able to use the plugin.

You will need to restart Excel after making changes to any of these options.

5.6. When starting, an error is received: "Could not load file or assembly 'System.Net.Http, Version=4.2.0.0'"

This error indicates that some of the components required for certain Microsoft applications to work on your PC are missing.

In this case, you must download and install the Microsoft .NET Framework components described in 2.2. System Requirements

Once you have installed it we suggest you restart your PC and try to launch raXL Stat.

5.7. When running the functions, I only see zeros "0"

In this case, it may be that there is an incorrect cell range reference in the formula or calculation, so we suggest selecting the correct cells according to the specification based on the requirement of the raXL Stat functions. It may also be that the license has expired.

5.8. raXL Stat ribbon menu in Excel not running or not responding

To fix and reactivate the ribbon menu in Excel, click on the same Help-Info menu and from there reset ribbon.

Finally, if you have any queries regarding the use of the raXL Stat plugin, are unable to fix a persistent problem, or are unable to find answers even after reading this manual, please feel free to contact us by sending an email to: rubenfapaza@gmail.com. The personal blog https://ruben-apaza.blogspot.com/p/raxl-stat.html will show updates, fixes to existing versions and releases of new versions.

Thank you!

Annexes

A. VBA Macro with Button

```
Option Explicit
Option Base 1
Sub Macro_GARCHCoeff()
Application.Calculation = xlCalculationManual
Dim runResult As Variant
Dim result() As Double
Dim rngRange As Range
Dim boolAscend As Boolean
Dim intP As Integer
Dim intQ As Integer
Dim intMuCond As Boolean
Dim i As Integer
Set rngRange = ActiveSheet.Range("B2:B1001")
boolAscend = False
intP = 1
intQ = 1
intMuCond = False
ReDim result(intP + intQ + 1, 1)
runResult = Application.Run("ra_GARCH_Coeff", rngRange, boolAscend, intP, intQ, intMuCond)
For i = 1 To intP + intQ + 1
result(i, 1) = runResult(i, 1)
Next i
Range("J3:J5").Select
Selection.FormulaArray = result
Application.Calculation = xlCalculationAutomatic
End Sub
```

B. Release Notes

November 15, 2024 - Version: 0.3 [Price $40/year]

- New: Calculate and graph the Value at Risk (VaR) of an investment portfolio.
- New: Perform retrospective testing with z-test and LR-Kupiec.
- New: Calculate the weights, risk and return of an investment portfolio.
- New: Calculate the minimum variance and tangency weights of an investment portfolio.
- New: Calculate and graph the Efficient Portfolio Frontier (EPF) and Capital Market Line (CML) of an investment portfolio.
- New: Run a simulation of an investment portfolio for the Efficient Frontier.
- Version 0.3 [Beta]: 76 Public Add-on User Defined Functions (UPFs).

October 22, 2024 - Version: 0.2.1

- Improved: Added frequency and density options to the histogram.

October 14, 2024 - Version: 0.2 [Price $ ~~30~~/year]

- New: Histograms with normal and cumulative curve.
- New: Boxplot with outliers.
- New: Normality test: Shapiro-Wilk, Anderson-Darling and Jarque-Bera tests.
- New: Descriptive statistics and normality test.
- New: Covariance matrix, coefficient correlation and correlation test.
- New: Missing data count.
- New: XY Scatterplot Matrix.
- Version 0.2 [Beta]: 50 additional public user-defined functions (UPFs).

September 30, 2024 - Version: 0.1.1.2

- Fixed: Dickey-Fuller regression sign with trend.
- Improved: GARCH model.

September 11, 2024 - Version: 0.1.1.1

- Fixed: 30-day trial version does not activate when launching .xll.

September 9, 2024 - Version: 0.1.1

- Fixed: Not showing decimals in GARCH model mu mean.
- Fixed: Not showing MaxLag in autocovariance function matrix.

August 29, 2024 - Version: 0.1 [Price $ ~~20 /year]~~

- Fixed: Release for licensed and trial versions.
- New: User Manual ES/EN.

August 28, 2024 - Version: 0.0 [Initial Release]

- Calculate and graph the autocorrelation function (ACF).

- Calculate and graph the partial autocorrelation function (PACF).
- Perform the white noise and independence test with the Ljung-Box or Box-Pierce test.
- Perform the Unit Root and Stationarity test with the Augmented Dickey-Fuller (ADF) test or the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test.
- Calculate the coefficients, estimate, forecast and graph the ARIMA(p,d,q) models, that is, AR(p), MA(q) and ARMA(p,q).
- Calculate the coefficients, estimate, forecast and graph the ARCH(p) and GARCH(p,q) models.
- Version 0.0 [Beta] features over 30 additional public User Defined Functions (UPFs).