Documentation of the Excel Add-In

RatingAnalytics

Version 1.3.3, December 2014

Table of Contents:

1 Introduction

This document describes the functionality of the Excel add-in Rating Analytics. It describes all functions contained in the add-in and how to call them and use them. It does not describe the mathematical details behind the functions. Everything can be found in Engelmann B. and R. Rauhmeier (eds.), 2011, The Basel II Risk Parameters: Estimation, Validation and Stress Testing – with Applications to Loan Risk Management, 2nd edition.

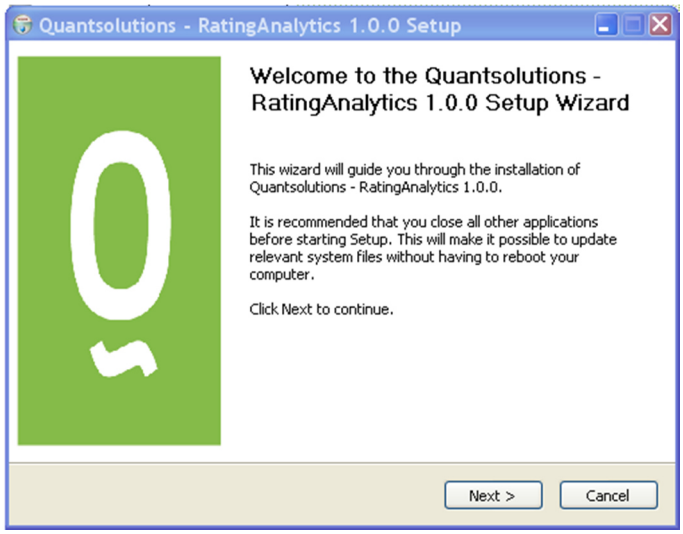
The Excel add-in contains the following functionality:

- Estimation of logistic regression models
- Measuring the discriminative power of ratings using ROC curves and AUROC
- Calculation of AUROC and its 95% confidence interval
- Statistical test for the difference in AUROC of two ratings
- Statistical tests on the accuracy of PD estimates
- Statistical test on the difference of PD estimates of two ratings
- Pluto-Tasche method for estimating default probabilities of low-default portfolios
- Auxiliary functions for data analysis

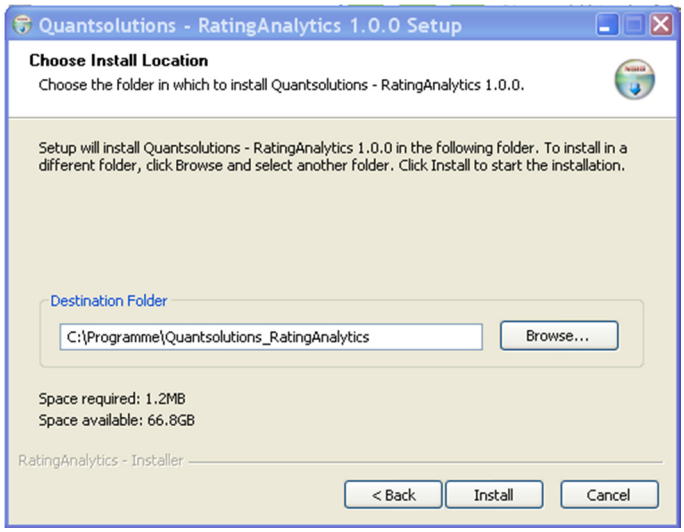
The document consists of two sections. In the first section it is explained how to use the add-in in Excel, i.e. how to install the add-in and how to call a function. The second section explains all the functions that are contained in the add-in. For every function it is explained what the function is doing, the list of its parameter list and the results it returns.

2 Installation and Using the Add-Ins in Excel

To start the installation process of the Excel add-in rating analytics, double-click the file RatingAnalytics.exe. The following window opens



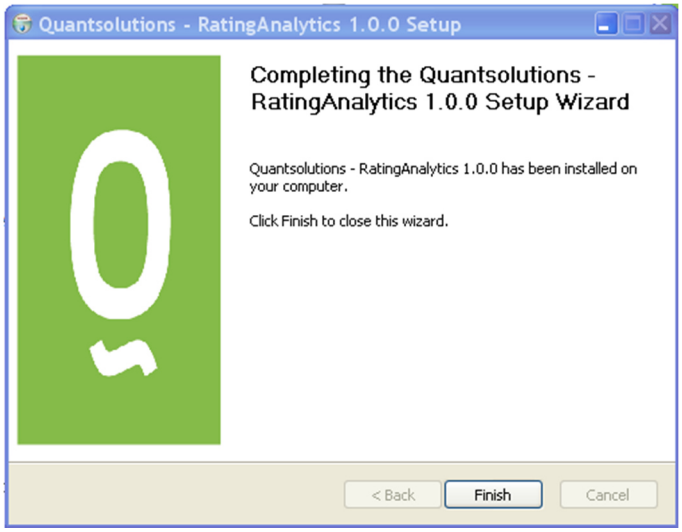
By clicking on the "Next" button you proceed to a screen where you can specify the installation directory



After you have specified the installation directory and after clicking the "Install" button, the following three files will be installed into the installation directory:

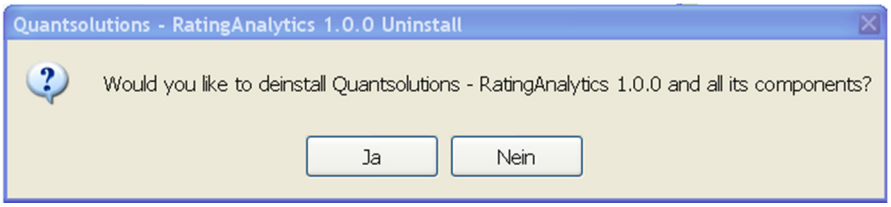
- Quantsolutions_RatingAnalytics.xls
- RatingAnalytics.xll
- uninst.exe

After a successful installation a confirmation window opens

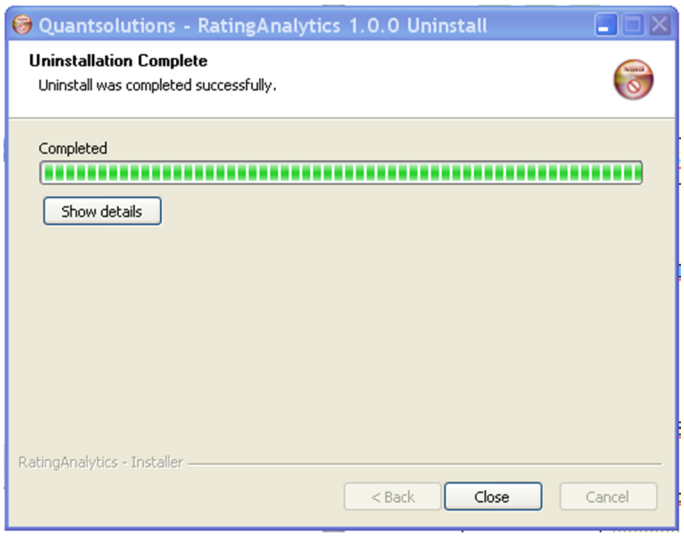


Confirm the installation by clicking the “Finish” button.

To de-install the add-in go into the installation directory and double-click the file uninst.exe



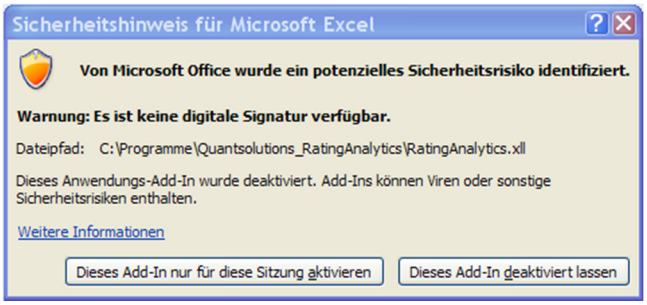
Confirm by clicking the “Yes” button. A confirmation window opens.



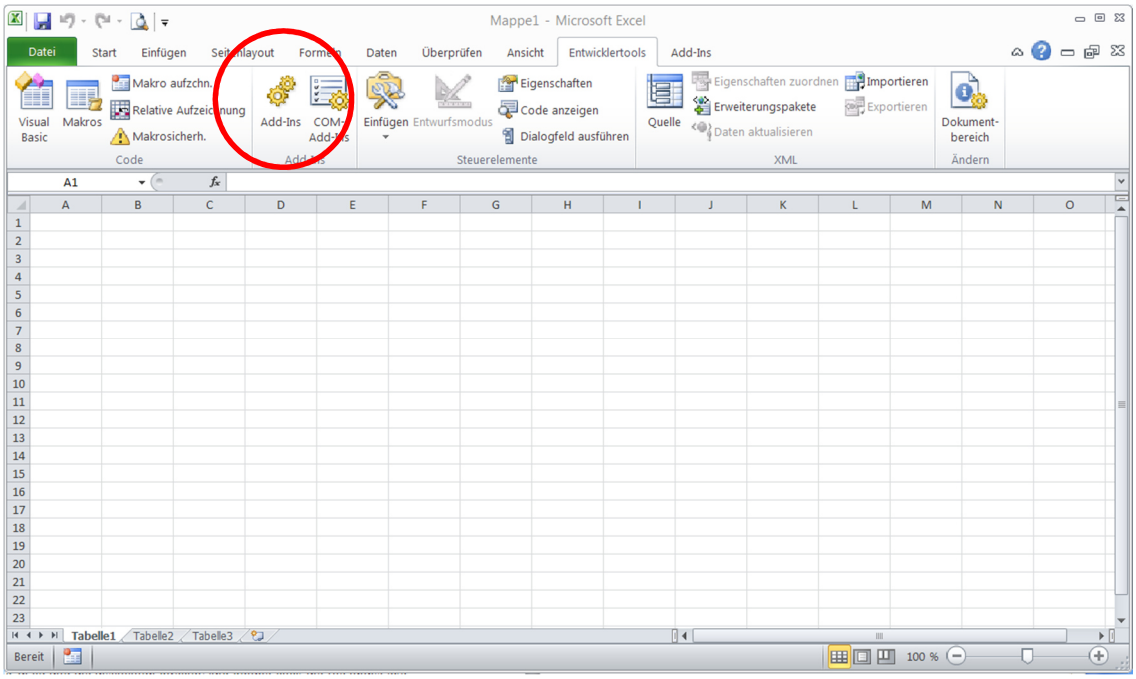
Close the window to finish the de-installation process.

2.1 Direct Use of RatingAnalytics.xll

To use the add-in double-click the file RatingAnalytics.xll. Excel opens together with a security window which contains a warning saying that this add-in might be a security risk (which is not the case, you can trust me ☺).

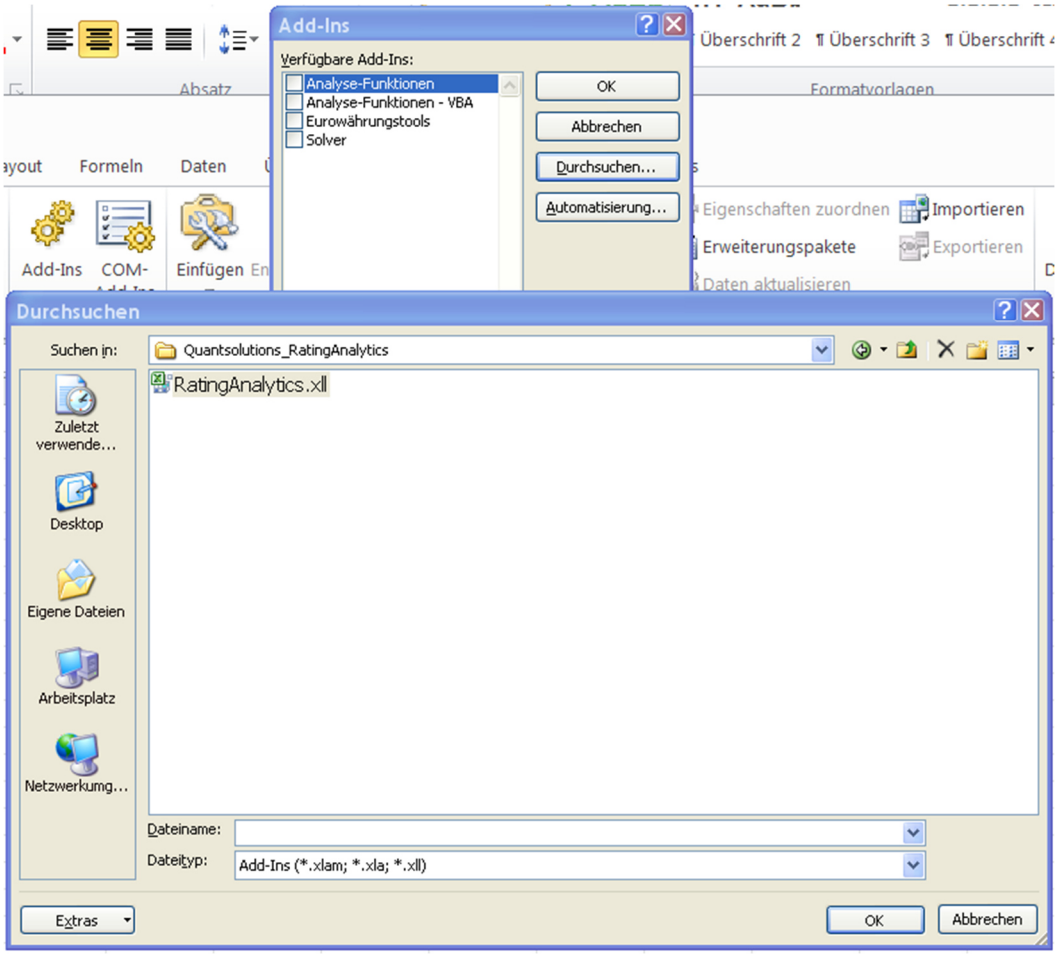


To activate the add-in click on "Activate the add-in for this session". If you wish to have the functionality of the RatingAnalytics add-in available whenever you open Microsoft Excel, open Excel, go to "Developer Tools" (which might be not visible immediately after installation, you have to activate it) and click on Add-Ins



A window showing all available add-ins opens. To install the RatingAnalytics add-in, click on search and go to the installation directory of the add-in. Click on the add-in and after that on "OK". This will install the add-in permanently. To remove the file from the add-ins that will be loaded automatically

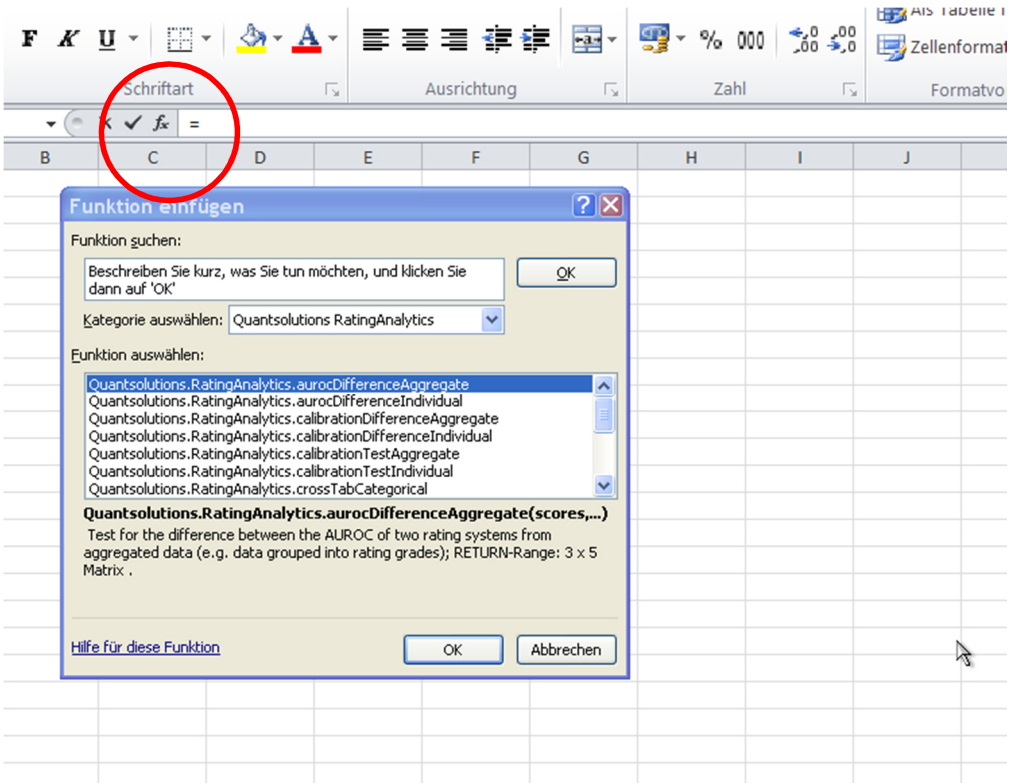
into Excel, the add-in has to be de-installed from the hard drive. When you open Excel after the de-installation it will ask for de-installation of the add-in from the list.



There might still be problems running the add-in if the security settings of Excel are too restrictive. The specific way to relax the security restriction of Excel depends on the version of Excel you are using. In any case you have to go to the menu “options” in Excel. Under Excel 2003 you have to go to the menu point “Security -> Macro Security” and decrease the security level to medium. For higher versions of Excel you have to go to the security centre, go to the security settings and add the installation directory to the trustworthy directories.

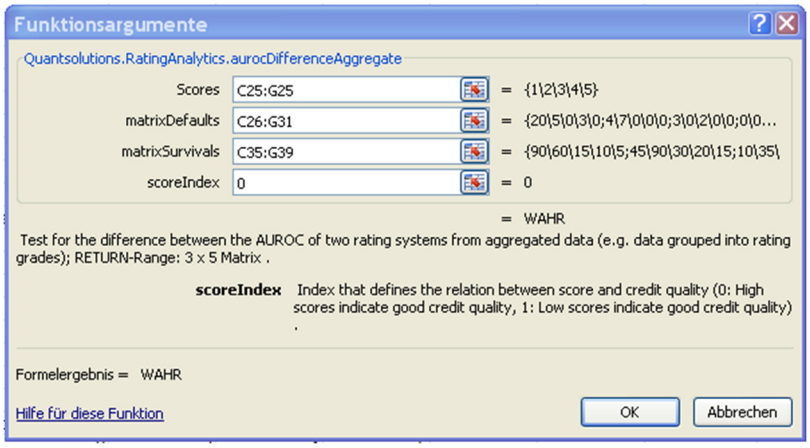
To check if the add-in is loaded into Excel, click into a cell and after that open the function assistant of Excel by clicking on the symbol in the red circle. A menu opens. Go into the functions menu and try to find the category “Quantsolutions RatingAnalytics”. If you can find it the add-in was loaded correctly.

If you do not find this category you have either forgotten to load the add-in or still a problem with your security settings.

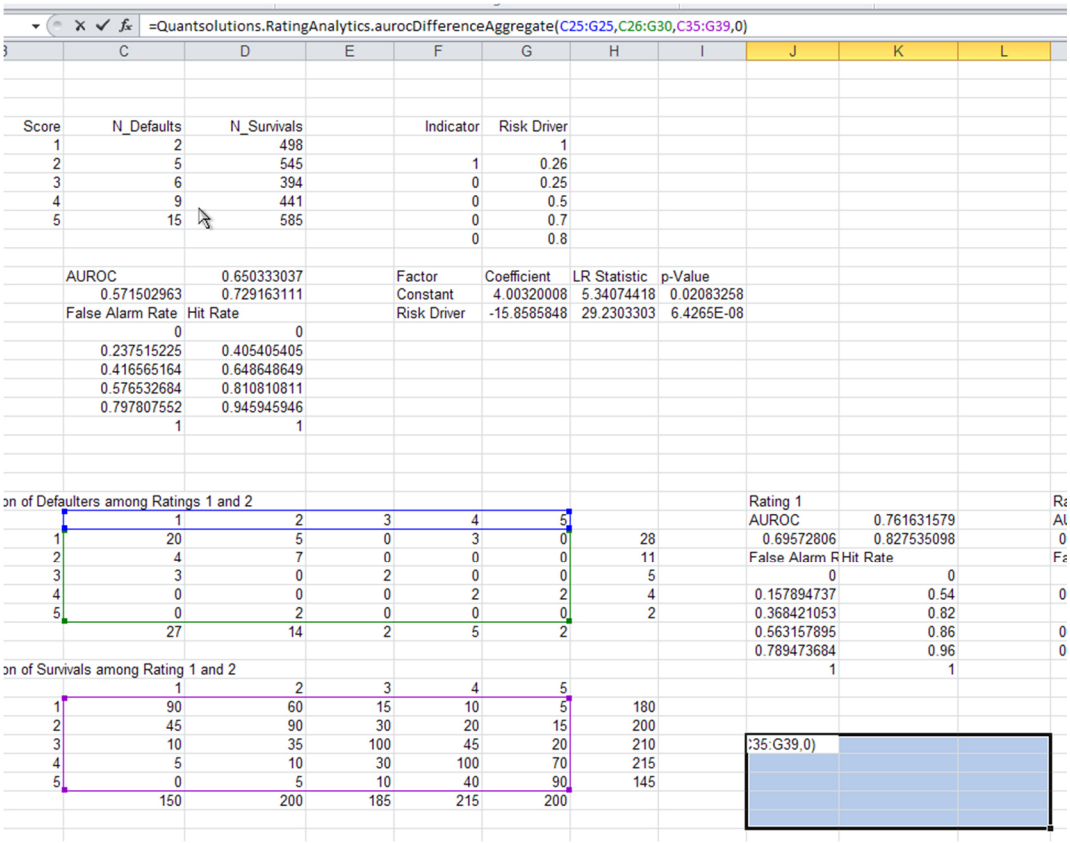


To see how to use the add-in, open the Excel sheet Quantsolutions_RatingAnalytics.xls where you can find examples for all the functions that are implemented in the add-in.

The function assistant will guide you in using the functions. After clicking on the name of the function you wish to use a dialog opens that describes its functionality, its input parameters and the results it will deliver. This is shown for the function auroDifferenceAggregate in the sequel.



After entering the input data and clicking “OK” you will see only one entry of the result matrix. This function resturns a 3 x 5 matrix as described in the function assistant. To see all results you have to mark the 3 x 5 matrix in the Excel sheet, enter the command line, and hit “Control + Shift + Enter”.

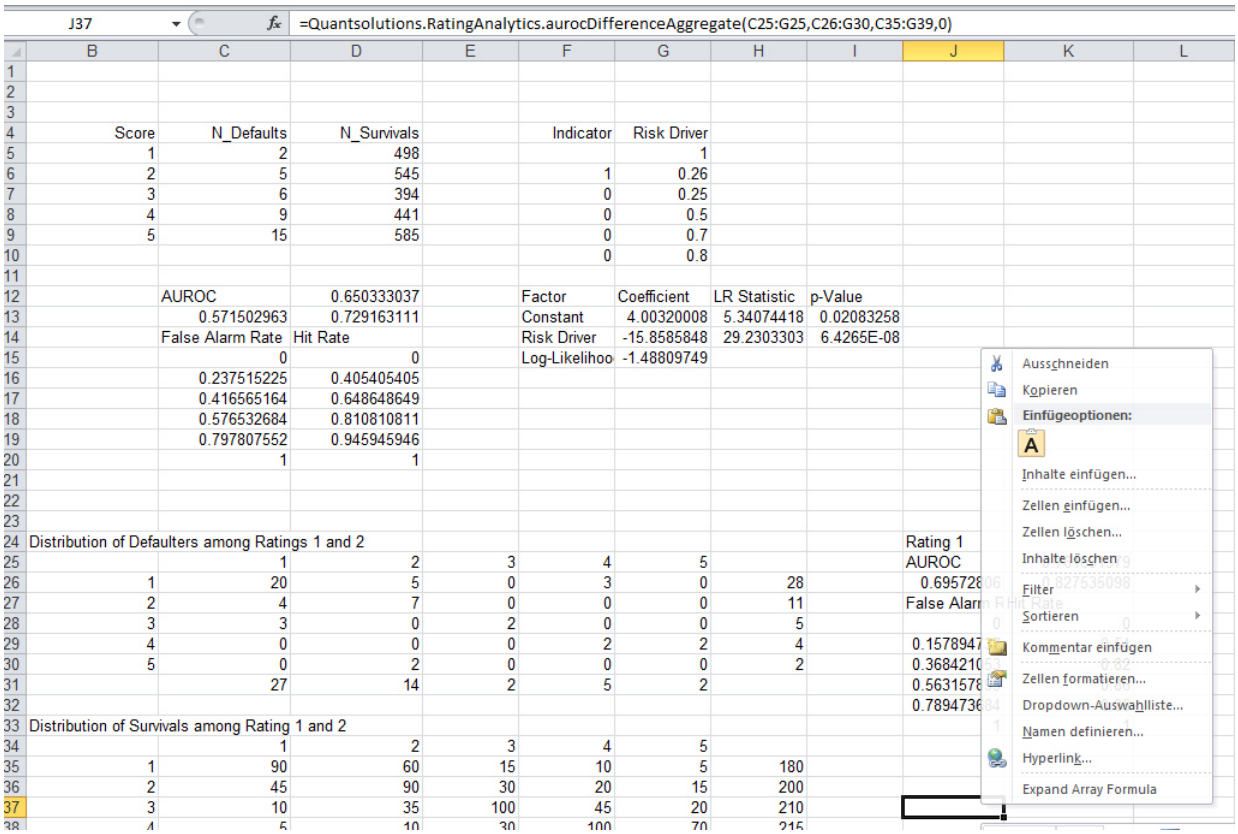


If you mark a smaller matrix you will see only parts of the results. If you mark a larger matrix Excel will fill the remaining space with the symbol “#NV”.

2.2 Alternative Use of RatingAnalytics.xla

A more user-friendly alternative for using this add-in is installing the file RatingAnalytics.xla instead of RatingAnalytics.xll. The add-in RatingAnalytics.xla hardly contains any functionality. It is just carrying out two operations:

- Automatically loading the add-in RatingAnalytics.xll
- Loading a VBA macro which automatically expands array formulas.



Instead of marking the output range of an array formula and hitting "Control+Shift+Enter" you can alternatively mark the cell which contains the command, click the right mouse button and move on "Expand Array Formula". This macro will automatically return the matrix containing the command's results. The "Expand Array Formula" macro works for all Excel commands that return more than one number, e.g. it could also be used for MMult.

3 Functions of the Add-In Rating Analytics

In this section every function that is contained in the add-in is described in detail. It is explained what the function is doing, what input parameters it expects, and what output it delivers.

3.1 The Routine adjustedPowerStat

This function computes a version of the power statistics for shadow ratings. Here, the input is not an indicator value for default/survival as for ROC curves but an external PD. The perfect rating system in this context gives a sorting of the external PDs in decreasing order. The closer the rating scores are in alignment with the perfect rating, the closer is the adjusted PowerStat to one.

Name	Quantsolutions.RatingAnalytics.adjustedPowerStat		
Parameters	Scores	Numeric Vector	Vector with numeric score values (e.g. PDs or enumerations) corresponding to each debtor
	shadowPds	Numeric Vector	Vector with external PDs corresponding to each debtor
	ScoreIndex	Integer	This index tells the function if high rating scores indicate good credit quality (ScoreIndex = 0) or if low rating scores are a sign of high credit quality (ScoreIndex = 1)
Return Values	This function returns a **3 x 2** matrix containing the adjusted power statistics for the rating model and the areas covered by the perfect rating and rating model.		

3.2 The Routine aurocDifferenceAggregate

In this function a test on the difference of the AUROC of two rating systems (Rating 1 and Rating 2) is carried out (the test is due to DeLong, DeLong and Clarke-Pearson). The null hypothesis is that the AUROC of both systems is identical. This function expects data of rating systems where debtors have been grouped into grades.

Name	Quantsolutions.RatingAnalytics.aurocDifferenceAggregate

Parameters	Scores	Numeric Vector	Vector with numeric score values (e.g. PDs or enumerations) for each rating grade
	MatrixDefaults	Integer Matrix	Integer valued matrix with dimension #grades x #grades that contains the information about defaults, e.g. the entry (2, 3) of this matrix contains the number of defaults that are assigned to grade 2 by Rating 1 and to grade 3 by Rating 2
	MatrixSurvivals	Integer Matrix	Analogous matrix to MatrixDefaults but contains the information about survivals
	ScoreIndex	Integer	This index tells the function if high rating scores indicate good credit quality (ScoreIndex = 0) or if low rating scores are a sign of high credit quality (ScoreIndex = 1)
Return Values	This function returns a **3 x 5** matrix containing the AUROC of both rating models, the value of the test statistics and the test's p-value.		

An example for this function can be found in the worksheet "Examples I".

3.3 The Routine aurocDifferenceIndividual

This function is analogous to the function aurocDifferenceAggregate. A statistical test on the difference of the AUROC of two rating systems is computed where in this case the data is not grouped into rating systems but each debtor has his individual rating (e.g. default probability). The outputs of the two functions are identical while the inputs of this function are vectors containing the ratings of each debtor in the two different rating systems.

Name	Quantsolutions.RatingAnalytics.aurocDifferenceIndividual		
Parameters	ScoresRating1	Numeric Vector	Vector with numeric score values (e.g. PDs or enumerations) for each individual debtor corresponding to rating system 1
	ScoresRating2	Numeric Vector	Vector with numeric score values (e.g. PDs or enumerations) for each individual debtor corresponding to rating system 2

	Indicators	0 / 1 Vector	Default Indicator (0 = Survival, 1 = Default) of each debtor
	ScoreIndex	Integer	This index tells the function if high rating scores indicate good credit quality (ScoreIndex = 0) or if low rating scores are a sign of high credit quality (ScoreIndex = 1).
Return Values	This function returns a **3 x 5** matrix containing the AUROC of both rating models, the value of the test statistics and the test's p-value.		

An example for this function can be found in the worksheet "Examples III".

3.4 The Routine calibrationDifferenceAggregate

This routine tests the difference in calibration of two rating systems where the output of each rating system is a default probability. The measure of calibration in this test (the test is known as Redelmeier test) is the mean squared error (MSE). The null hypothesis is that the mean squared error of both rating systems is equal. The function is written for data of rating systems where debtors are already grouped to rating grades using the same master scale for each rating system. It returns the MSE of each rating system, the Redelmeier test statistic and its p-value.

Name	Quantsolutions.RatingAnalytics.calibrationDifferenceAggregate		
Parameters	DefaultProbabilities	Numeric Vector	Vector with numeric score values (e.g. PDs or enumerations) for each individual debtor corresponding to rating system 1
	MatrixDefaults	Integer Matrix	Integer valued matrix with dimension #grades x #grades that contains the information about defaults, e.g. the entry (2, 3) of this matrix contains the number of defaults that are assigned to grade 2 by Rating 1 and to grade 3 by Rating 2
	MatrixSurvivals	Integer Matrix	Analogous matrix to MatrixDefaults but contains the information about survivals

Return Values	This function returns a **3 x 5** matrix containing the MSE of both rating models, the value of the Redelmeier test statistics and the test's p-value.

An example for this function can be found in the worksheet "Examples VII".

3.5 The Routine calibrationDifferenceIndividual

This routine is the analogue to the routine calibrationDifferenceAggregate. Instead of data from rating systems it expects a vector of individual PDs corresponding to the two rating systems.

Name	Quantsolutions.RatingAnalytics.calibrationDifferenceIndividual		
Parameters	DefaultProbRating1	Numeric Vector	Vector with default probabilities for each individual debtor corresponding to rating system 1
	DefaultProbRating2	Numeric Vector	Vector with default probabilities for each individual debtor corresponding to rating system 2
	Indicators	0 / 1 Vector	Default Indicator (0 = Survival, 1 = Default) of each debtor
Return Values	This function returns a **3 x 5** matrix containing the MSE of both rating models, the value of the Redelmeier test statistics and the test's p-value.		

An example for this function can be found in the worksheet "Examples VII".

3.6 The Routine calibrationTestAggregate

This function tests whether a rating system is well calibrated. It returns the results of two tests. The first test is known as the Spiegelhalter test. It uses the mean squared error (MSE) as a measure for calibration. The null hypothesis is that the MSE of a rating system equals its expected value. The second test is known as Hosmer-Lemeshow test. It measures the difference of default probabilities and realized default rates over all grades of a rating system. For both tests the test statistics and the corresponding p-values are computed.

Name	Quantsolutions.RatingAnalytics.calibrationTestAggregate

Parameters	DefaultProbabilities	Numeric Vector	Vector with default probabilities corresponding to each rating grade of a rating system
	NumDefaults	Integer Vector	Vector with the number of defaults corresponding to each rating grade
	NumSurvivals	Integer Vector	Vector with the number of survivals corresponding to each rating grade
Return Values	This function returns a **2 x 8** matrix containing the test statistics corresponding to the two tests, some intermediate results, and the p-values of both tests.		

An example for this function can be found in the worksheet "Examples VI".

3.7 The Routine calibrationTestIndividual

This routine is the analogue to calibrationTestAggregate. In this case there is a difference to the corresponding aggregate-function because the Hosmer-Lemeshow test does not work for individual data. This means that only the Spiegelhalter test is carried out in this case.

Name	Quantsolutions.RatingAnalytics.calibrationTestIndividual		
Parameters	DefaultProbabilities	Numeric Vector	Vector with default probabilities of each individual debtor of the rating system
	Indicators	0 / 1 Vector	Default Indicator (0 = Survival, 1 = Default) of each debtor
Return Values	This function returns a **2 x 6** matrix containing the test statistics corresponding to the Spiegelhalter test, some intermediate results, and the p-value.		

An example for this function can be found in the worksheet "Examples VI".

3.8 The Routine crossTabCategorical

This function is a tool to analyse categorical data during the development process of a rating model or score card. If for each debtor the value of a categorical risk driver is known together with the default indicator, this function can be used to compute a cross tabular, i.e. a table which gives the number of defaults and survivals corresponding to each category together with its Weight of Evidence.

Name	Quantsolutions.RatingAnalytics.crossTabCategorical		
Parameters	Categories	General Vector	Vector with the realizations of a categorical risk driver for each individual debtor
	Indicators	0 / 1 Vector	Default Indicator (0 = Survival, 1 = Default) of each debtor
Return Values	This function returns a **(#categories + 1) x 4** matrix containing the number of default, the number of survivals and the Weight of Evidence corresponding to each debtor. (#categories means the number of different categories)		

An example for this function can be found in the worksheet "Examples IV".

3.9 The Routine crossTabNumerical

This routine is the analogue of crossTabCategorical for numerical risk drivers. Here, the user has to specify the number of categories and the routine will automatically generate roughly equally distributed categories using the desired number and display the score boundaries. Alternatively, intervals of the numerical values which should define a single category can be defined by the user explicitly. The convention applied it that upper end points of each interval belong to the interval while the lower boundaries are excluded. To be consistent with this convention it is required that upper boundaries of categories and lower boundaries of succeeding categories are identical.

Name	Quantsolutions.RatingAnalytics.crossTabNumerical		
Parameters	Scores	Numeric Vector	Vector with the realizations of a numerical risk driver for each individual debtor (can be any score value, not necessarily a default probability)
	Indicators	0 / 1 Vector	Default Indicator (0 = Survival, 1 = Default) of each debtor
	NumberCategories	Integer	Number of categories among which the borrowers should be distributed according to their score values
	LowerBoundaries	Numeric Vector	Vector with values of the lower boundaries of each interval defining a category

	UpperBoundaries	Numeric Vector	Vector with values of the upper boundaries of each interval defining a category
Return Values	This function returns a **(#categories + 1) x 6** matrix containing the number of default, the number of survivals, the median score values of each category, and the Weight of Evidence corresponding to each debtor. (#categories means the number of different categories)		

An example for this function can be found in the worksheet "Examples V".

3.10 The Routine getInfo

This is a function without arguments containing copyright information.

Name	Quantsolutions.RatingAnalytics.getInfo
Parameters	This function does not have any arguments
Return Values	Copyright information of this Excel Add-In

This function is used in the worksheet "Examples I".

3.11 The Routine getVersion

This is a function without arguments containing the version information.

Name	Quantsolutions.RatingAnalytics.getVersion
Parameters	This function does not have any arguments
Return Values	Version of this Excel Add-In

This function is used in the worksheet "Examples I".

3.12 The Routine logisticRegression

This function computes the coefficients of a logistic regression by maximum likelihood estimation. In addition a statistical test on the significance of the coefficients is performed. The specific test that is

implemented is the likelihood ratio test. For each coefficient of the model equation the test statistic and the p-value of the likelihood ratio test are calculated.

Depending on the input data you are using, this function might not always be come up with a solution of the maximum likelihood regression. If it does not find a solution, the function will return the error message "Newton solver of the logistic regression did not converge". This might be the case, for instance, if your data contains perfect separation, i.e. that a combination of the risk drivers can be found that leads to a perfect rating system that separates all defaults from the survivals in the data.

Name	Quantsolutions.RatingAnalytics.logisticRegression		
Parameters	FactorNames	String Vector	This vector includes a name (string) for each factor included in the analysis. This is needed to identify the results corresponding to each factor in the results matrix.
	FactorIndices	0 / 1 Vector	This vector is needed to define the factors that should be included in the model estimation (0 = factor not included, 1 = factor included). This allows the estimation of different models by just changing these indices.
	FactorValues	Numeric Matrix	Matrix containing the values of all factors included in the analysis
	Indicators	0 / 1 Vector	Default Indicator (0 = Survival, 1 = Default) of each debtor
Return Values	This function returns a **(#factors + 3) x 4** matrix containing the names of each factor included in the model estimation, the value of the corresponding model coefficient, the value of the likelihood ratio test statistic and the corresponding p-value. (#factors means the number of factors included in the model estimation)		

An example for this function can be found in the worksheet "Examples I".

3.13 The Routine lowDefaultPortfolioPD

This function computes default probabilities of low-default portfolios following the method developed by Katja Pluto and Dirk Tasche. A reference of this method is Chapter 5 in Engelmann / Rauhmeier (eds.), The Basel II Risk Parameters – Estimation, Validation, Stress Testing – with Applications to Loan Risk Management, Springer, 2011.

Name	Quantsolutions.RatingAnalytics.lowDefaultPortfolioPD		
Parameters	NumDefaults	Integer Vector	This vector contains the observed number of defaults for each rating grade for which a default probability should be estimated.
	NumSurvivals	Integer Vector	This vector contains the observed number of survivals for each rating grade for which a default probability should be estimated.
	Correlation	Double	Mean asset correlation that is used to compute the distributions of the Pluto-Tasche method.
	Quantile	Double	Quantile of the distributions of defaults that is used for computing PDs in the Pluto-Tasche method
	AveragePortfolioPD	Double	Average PD of the total portfolio needed for re-scaling the estimated PDs.
Return Values	This function returns a vector of length "number of rating grades" containing the estimated PDs of each rating grade.		

An example for this function can be found in the worksheet "Examples VIII".

3.14 The Routine mapCategoricals2Scores

This routine is an auxiliary function to help preparing data for a logistic regression. Suppose you have computed a cross tabular using the routine crossTabCategorical and wish to use the calculated Weight of Evidence values as inputs to a logistic regression. To do so you have to map back the Weight of Evidence values to the individual debtors. This can be done by using this function.

Name	Quantsolutions.RatingAnalytics.mapCategoricals2Scores		
Parameters	CategoricalScores	General Vector	Vector containing the categorical scores of each debtor
	CategoricalSet	General Vector	Set of possible categorical score values
	ScoreValueSet	Numeric Vector	Vector of numerical scores (e.g. Weight of Evidence values) corresponding to each categorical score

Return Values	Vector of a dimension equal to the number of debtors. It contains for each debtor the numerical score value corresponding to his categorical score value.

An example for this function can be found in the worksheet “Examples IV”.

3.15 The Routine mapNumericals2Scores

This function is the analogue to the routine mapCategoricals2Scores. It maps the output of a cross tabular for numerical score values from the function crossTabNumerical back to each debtor.

Name	Quantsolutions.RatingAnalytics.mapNumericals2Scores		
Parameters	NumericalScores	Numeric Vector	Vector containing the numerical scores of each debtor
	ScoreValueSet	Numeric Vector	Vector of numerical scores (e.g. Weight of Evidence values) corresponding to each numerical score range
	LowerBoundaries	Numeric Vector	Vector with values of the lower boundaries of each interval defining a category
	UpperBoundaries	Numeric Vector	Vector with values of the upper boundaries of each interval defining a category
Return Values	Vector of a dimension equal to the number of debtors. It contains for each debtor the numerical score value corresponding to his numerical score category.		

An example for this function can be found on the worksheet “Examples V”.

3.16 The Routine rocCurveAggregate

This function calculates the ROC curve in a format that can be directly used for plotting the ROC curve in Excel. In addition it computes the area below the ROC curve together with the 95% confidence interval of the area below the ROC curve. This routine expects data of rating systems where debtors are grouped into grades.

Name	Quantsolutions.RatingAnalytics.rocCurveAggregate

Parameters	Scores	Numeric Vector	Vector with numeric score values (e.g. PDs or enumerations) for each rating grade
	NumDefaults	Integer Vector	Integer valued vector with the number of defaults corresponding to each rating grade
	NumSurvivals	Integer Vector	Integer valued vector with the number of survivals corresponding to each rating grade
	ScoreIndex	Integer Value	This index tells the function if high rating scores indicate good credit quality (ScoreIndex = 0) or if low rating scores are a sign of high credit quality (ScoreIndex = 1)
	OutputIndex	Integer Value	Controls the output (0 = only the area below the ROC curve and the 95% confidence interval are computed, 1 = in addition the full ROC is calculated)
Return Values	If the value of OutputIndex is 0 the function returns a **2 x 2** matrix containing the area below the ROC curve and its 95% confidence interval. For OutputIndex = 1, the return matrix has the dimension **(#grades + 3) x 2** containing the ROC curve in addition. (#grades is the number of rating grades)		

An example for this function can be found in the worksheet “Examples I”.

3.17 The Routine rocCurveIndividual

This function is analogous to the function rocCurveAggregate. The outputs of the two functions are identical while the inputs of this function are vectors containing data corresponding to each individual debtor.

Name	Quantsolutions.RatingAnalytics.rocCurveIndividual		
Parameters	Scores	Numeric Vector	Vector with numeric score values (e.g. PDs or enumerations) for each rating grade
	Indicators	0 / 1 Vector	Default Indicator (0 = Survival, 1 = Default) of each debtor
	ScoreIndex	Integer Value	This index tells the function if high rating scores indicate good credit quality

			(ScoreIndex = 0) or if low rating scores are a sign of high credit quality (ScoreIndex = 1)
	OutputIndex	Integer Value	Controls the output (0 = only the area below the ROC curve and the 95% confidence interval are computed, 1 = in addition the full ROC is calculated)
	NumBuckets	Integer Value	Defines the number of buckets that should be used in the calculation of the full ROC curve (only relevant for OutputIndex = 1)
Return Values	If the value of OutputIndex is 0 the function returns a **2 x 2** matrix containing the area below the ROC curve and its 95% confidence interval. For OutputIndex = 1, the return matrix has the dimension **(NumBuckets + 3) x 2** containing the ROC curve in addition.		

An example for this function can be found in the worksheet “Examples II”.