System Requirements

Before installing the Causal Analytics Toolkit (CAT) on Microsoft Windows, the following must be installed:

- Microsoft Excel 2007 or later
- R version 3.2.2 or later

In short, as long as you have Microsoft Windows, and you can run Excel 2007 or later (either 32-bits or 64-bits), then you can use CAT.

Install the “Causal Analytics Toolkit”

Double-click setup.exe, then follow the on-screen instructions. If you already have R installed, the installer will check whether the installed version can be used by CAT (32-bit R must be installed). If you have not installed R, then the installer will download and install R version 3.3.1 automatically. Do not select Python Plugin unless you have received instructions from tcoxdenver@aol.com.

When installing R, options 32-bit Files and Save version number in the registry are checked by default (see below). Do not uncheck these, otherwise CAT will not work.

The CAT installer also provides an option to install R libraries (see Figure 1 above). It is recommended to check this box when you install CAT the first time. When checked, this will install all the libraries used by
CAT in folder Documents/R/win-library/3.x where 3.x represents the first two digits of the R version installed (currently 3.2 or 3.3). If you do not want to install packages this way, you need to manually install the required packages: a utility function is provided for this purpose (discussed below).

When installation is complete, an Excel add-in “Causal Analytics Toolkit.xlam” will be installed in the default Excel add-in folder. (The target folder can be changed during the installation process if desired). The final screen for the installation process should look as follows:

Note: If you manually changed R between version 3.2.* and 3.3.* after CAT is installed, then you must reinstall CAT again.

Set up the “Causal Analytics Toolkit” Excel add in

Click the top left office icon in Excel (the “Office Button”) then click the “Excel Options” button. Select “Add-ins”, then click the “Go...” button next to Manage Excel Add-ins. Select (check) the “Causal Analytics Toolkit.”

When the “Causal Analytics Toolkit” appears in the Excel ribbon as shown below, installation is complete.

The Causal Analytics Toolkit can be uninstalled at anytime using the standard Programs and Features in Windows’ control panel.
To confirm that CAT is running at any time, double-click on any cell in an Excel workbook and enter a simple arithmetic command such as “R: 2 + 3” then hit the RETURN key. The $R$: prefix alerts CAT that what follows next is an R command. If CAT is running, it will return the R result “[1] 5” on the next line, and the cell with $R$: command should become bold. If CAT is not running, then closing Excel and reopening it will re-start CAT.

![R: 2 + 3](image)

**Install R Packages**

CAT uses several dozen R packages; these must be installed while the computer is connected to the internet. It is recommended to use the CAT installer to install these packages. But if choose to install R packages manually, follow these steps:

- Start Excel and open a new workbook. (It is important that no existing worksheet named *Utilities* exists)
- From the CAT ribbon, select the *Packages* dropdown menu (in the *Methods* section of the CAT ribbon). Select *Utilities* from that menu. CAT will create a new worksheet called *Utilities*.
- In the *Utilities* worksheet, select column B (by clicking on the column heading to highlight the column), or select any subset of packages in column B, then click on *Run R script* (in the *Methods* section of the CAT ribbon).

*Note:* Installing packages may take several minutes, depending on your computer and internet connection speed.

![Running R packages](image)

*Note:* During package installation, dialog boxes may pop up to prompt user input. Some dialog boxes may be hidden behind the Excel window, so watch the Windows’ taskbar for flashing icons that need attention. Click on “Yes” or “Next” or “Ok” as needed to install all packages.

**Using CAT**

The CAT Excel add-in provides relatively simple, powerful commands and a point-and-click interface for doing advanced analytics from Excel using R packages, even if the user does not know R. It can be used in many ways, from push-button, fully automated analyses to programming in R, depending on the level of user experience and familiarity with statistics and with R. For users who have no knowledge of R, a
few mouse clicks will display results from advanced R packages without the need to learn R or to know about the packages being used.

**Example: Push-Button Regression Modeling and Bayesian Network Modeling in CAT**

To illustrate CAT’s push-button analytics, open a new Excel workbook and select Sample1 from the Excel to R drop-down menu at the far left of the CAT ribbon. A new worksheet named Data will be created that looks like this:

![Excel Data Table]

Click anywhere in the spreadsheet to make sure only a single cell is selected, then click on the big Excel to R icon to send all columns to R; click on “Yes” when the message box appears, and click on “OK” to have CAT create an R data frame.

Highlight columns D-H (by clicking and swiping on the Excel column letters D-H). Click on the Automatic button in the Regression Models area of the CAT ribbon, as shown below:

![Regression Models in CAT]

CAT will then automatically select appropriate families of regression models, fit the models to the data, and display the results. The first column selected, column D (AllCause75) is treated as the dependent variable and the remaining variables highlighted are treated as candidate predictors (i.e., independent variables). CAT generates a new Regression tab with extensive output from the regression modeling, beginning with the following display:
This shows the regression coefficients for a quasi-Poisson regression model fit to the data. Additional outputs include 95% confidence intervals, Added-Variable plots showing how the dependent variable is predicted to change as each predictor is varied (i.e., assigned a range of counterfactual values, while holding all other variables at their actual values), importance measures, results from non-parametric (Random Forest) and linear regression modeling, and a plot showing which variables are used in linear regression models of increasing size that explain increasing proportions of the variance in the dependent variable. These outputs are generated by appropriate R packages and use the same format as these packages; thus, the modeling and interpretation of results can be studied in detail by using the extensive existing documentation on R packages. (To see which specific R packages are used by each CAT function, select View CAT Functions under the Function Builder drop-down menu in the Methods section of the CAT ribbon.) To generate a Bayesian Network for the same data, simply click on the Bayesian Network button (under the Causal Models section of the CAT ribbon). CAT will then generate a new tab called Bayesian that shows a Bayesian network (BN) structure for the data. (In such a BN diagram, nodes represent variables and arrows between nodes show that they are not statistically independent of each other, i.e., observing the value of one variable provides information about the value of the other.) CAT does not teach the user how to read and interpret its outputs, leaving this for the R package documentation. But it does enable users to generate a rich set of advanced statistical outputs with minimal effort, and with no knowledge of R or R packages, by selecting columns in an Excel spreadsheet and clicking on analytics buttons.
For intermediate users who wish to learn or extend their knowledge of R or CAT commands, CAT also automatically generates and displays the relevant commands (e.g., `CAT_regression(AllCause75, PM2.5, tmin, tmax, MAXRH)` for the above regression modeling, or `CAT_bnLearn(AllCause75, PM2.5, tmin, tmax, MAXRH)` to generate the Bayesian network). Typing such commands directly into Excel (prefixed by `R:` as in `R: CAT_regression(AllCause75, PM2.5, tmin, tmax, MAXRH)` for the above regression modeling, or `R: CAT_bnLearn(AllCause75, PM2.5, tmin, tmax, MAXRH)` to generate the Bayesian network) provides another way to run them without using the CAT ribbon interface. For advanced R users, any R or CAT formula can be entered into an Excel cell and then run as if the Excel cell were an R console. (Thus, typing `R: mean(tmin)` into an empty cell in the above example will return 50.3525 as the mean of the variable `tmin`.) In addition, CAT provides a Function Builder (the left-most option in the Methods section of the CAT ribbon) that guides users in selecting R package and CAT functions and populating them with appropriate argument values. Advanced users can also write an R script (a sequence of commands) in any Excel range, highlight the script, and then run it using CAT’s `Run R script` command from the CAT ribbon (at the bottom of the Methods section) to see results directly in the Excel spreadsheet.

**Basic Use Cases**

*Step 1. Create Data*

Although not strictly required, it is recommended practice to use *one data set per Excel workbook*, and to name that data sheet *Data*. (If you already have a data set open in a worksheet named *Data*, then skip this step and go to Step 2.)

Two sample data sets, Sample1 and Sample2, are included with the base CAT installation. These can be loaded from the *Excel to R* drop-down menu at the upper left of the CAT ribbon. Sample1 is a time series data set for daily fine particulate matter (PM2.5) concentrations, mortality counts among people at least 75 years old (AllCause75), and meteorological variables (daily minimum and maximum temperatures and maximum relative humidity) for the Los Angeles air basin. These data were kindly provided by Dr. Stan Young. Sample2 is a larger cross-sectional data set assembled from EPA and CDC (BRFSS survey) data in the public domain. To load a sample data set, simply click on it in the *Data Samples* menu.

Upon clicking Sample1 or Sample2, if an open *Data* sheet already exists, then it will be just activated, and no further changes will be made. Otherwise, a new worksheet *Data* will be created.

To apply CAT to a new data set, open a new workbook in Excel and create an Excel spreadsheet named *Data* that contains the data set. Variable names should be in the first (top) row. Data should start from the first column, and fill a range without any blank columns in middle.

**Note:** The current base version of CAT does not allow missing data. If needed, click on Clean Rows under the Split Column drop-down menu in the Data section of the CAT ribbon to delete all rows with missing data. Most R packages and CAT functions can be configured to work with missing data, e.g., using various imputation options, but the current base version of CAT assumes no missing data. If there are many missing data values and they are not missing at random, then the missing data may bias results.
**Step 2. Export Data to R**

Select some or all columns of the Data sheet (by clicking on the letter that Excel provides at the top of a column; click-and-dragging across multiple contiguous columns to multi-select them; and using Ctrl-Left-Click to select additional columns as desired, and Ctrl-Right-Click to deselect a column). Then, click the Excel to R button at the far left of the CAT ribbon to export the selected data columns into R. If no columns are selected, and only a single cell is selected, then all columns in the Data sheet will be export to R by default.

*Note for R users:* Each column is exported as a vector or factor depending on whether it is numeric or text. When numbers are stored as text in Excel (e.g., zip codes), CAT will export such text columns into R as factor. When columns with numbers only need to be treated as factor in R, e.g., Year or Age, format the column as text (see below): Excel will automatically set the column alignment to the left.

After the selected columns have been exported to R, CAT will prompt the user about whether to also export the entire collection of columns as an R data frame. It is recommended to select Yes and to keep the R data frame named Data. Accepting these defaults will create a named array of data (the data frame, named Data) that can be used by R programmers and by R and CAT. (For example, the data frame with name Data is used by the Data Explorer tool, discussed later, to display summaries of data in the Data sheet.) If your data set is very large, and you do not plan to use data frame in your R functions, there is no need to create a data frame from the columns.

When a column of data is exported into R, the column header will be used as the name of the R vector or factor. Special characters in the column header are replaced by _ in the R name. For example, column header “Test Header” results in “Test_Header” in R.

**Step 3. Verify R Data**

To verify that all/selected columns are exported into R, use the R to Excel menu. If a column header is shown in the list, then the column has been successfully exported to R. Any variable in the list can be imported back into Excel. (If the R variable is a vector or factor, there is an option to export it as a row or column.)
Step 4. Viewing Statistical Patterns with Data Explorer

CAT provides a unique Data Explorer feature that lets users view many statistical characterizations of the data simply by mousing over columns of data (or, for some more sophisticated analyses, by selecting one or more columns and then mousing over others). To activate it, just click on the Data Explorer icon (at the right of the Data section of the CAT ribbon). A new window will pop up (possibly after a brief pause). By default, this Data Explorer window will move with the mouse. As it moves, different results are displayed. The motion of the window can be controlled using the pin icon at the upper right corner of the window, which looks like this: ⬇️. To stop the explorer window from moving, double-click on any cell of Excel worksheet so the pin icon becomes down 🔔. Clicking on the pin icon itself will also toggle window movement on or off.

While the data explorer window is active, it will show results for the column under the mouse cursor. To freeze contents in the explorer widow, click this icon 🗝️ in the upper right corner of the explorer window, or right-click on any Excel worksheet cell that is not empty: the icon will be changed to 🔔. Clicking on the pin icon itself will also toggle window movement on or off.

When contents are frozen in the Data Explorer window, right-clicking any empty cell of an Excel worksheet will transfer contents (text or picture) from the Data Explorer window into Excel. This makes results viewed in Data Explorer available to paste into other applications such as MS Word or Power Point.

When the Data Explorer window is visible, moving the mouse cursor to a column shows statistical results, such as histograms, box plots, and descriptive statistics summary tables, for that column. Moving the mouse cursor up and down within a column shows different results for that column.

The content of the Data Explorer window may be a text table or a picture. If it is text, then double-clicking anywhere in the explorer table will pop up a table of options for which displays to include in the Data Explorer window. This options table can be also activated using the Options menu under the Data Explorer drop-down menu when the Data Explorer window is not visible. For large data sets, some
results can take up to several minutes to calculate. The *Options* menu (under the Data Explorer drop-down menu) allows the user to select which calculations to perform and which to skip.

Some analyses require the user to specify multiple columns. For example, correlations, as well as more advanced analytics options such as Bayesian Networks, require specifying the columns that are to be included in the analysis. Therefore, CAT allows the user to select multiple columns before activating *Data Explorer*. Doing so causes appropriate (multi-variable) displays to be generated as the mouse is moved. For example, clicking on one column and then passing the *Data Explorer* window over other columns will allow a variety of correlations to be viewed between the pre-selected column and the columns that the window (or cursor) passes over.

While the mouse is moving, the current column and all columns selected before the explorer was activated are highlighted. If highlighting is lost, just left single-click on any Excel cell to re-set the focus for Excel.

### Step 5. Using CAT models

To see results from a CAT model, select some columns from Data worksheet, then click the desired analytics button(s) or icons on the ribbon. The selected columns in the Data sheet are remembered, so you can click on each analytics option in the CAT ribbon (e.g., P for Poisson regression, B for Bayesian Network, etc.) after columns in the Data sheet have been selected. Results from running a CAT analytics model are placed into a sheet with a name corresponding to that analysis (e.g., “Correlations” for the results of correlation analyses, “Bayesian” for Bayesian Network, “Poisson” for a (quasi-)Poisson regression analysis, and so forth). The output sheet is cleared and repopulated each time a CAT analytics model is run. To save the results, just rename the output sheet to a sheet with a different name.

To run a CAT analytics model on some Excel columns, the Excel columns must have been already loaded into R using *Excel to R*. Three exceptions are *Mutual Info*, *Granger Test*, and *Transfer Entropies*. For these three analyses, the columns do not have to exist in R, and the *Data* sheet must be the current active sheet. (As usual, explanation and interpretation of these analyses is provided in the R documentation and the references therein; CAT simply makes it easy to apply them to data.)
**Advanced Use Cases**

For advanced R users, CAT can be used as an R console, with extra added conveniences including:

- Data in Excel worksheets and in R can be exchanged easily
- R commands can be entered into any cell and executed from Excel
- R output results can be placed in any selected cells

Any selected range in an Excel sheet can be exported into R using the *Excel to R* menu. Enter any valid R object name when prompted. If the selected range is 1-dimensional, then the range will be exported into R as a vector or factor depending on whether all values are numeric. If the column contains all numbers but formatted as Text, then it is exported into R as factor. If the range is 2-dimensional, then the range will be exported into R as a data frame. CAT will prompt for whether the top row of a 2-dimensional selection should be used as header.

**Note:** *If a data range is modified after it is exported into R, the data in R will not be automatically updated. To update data in R, do “Excel to R” again.*

For the following example, there are two rows. First select the top row (yellow), then click *Excel to R* (top left of the CAT ribbon). When prompted, type *a*. This is equivalent to the R command: \( a \leftarrow c(10, 20, 50, 60) \). Similarly, select the second row (green), and export it to vector *b* in R: this is equivalent to the R call \( b \leftarrow c(15, 35, 45, 25) \). Note: the selection may be also a 1-dimensional column.

<table>
<thead>
<tr>
<th></th>
<th>10</th>
<th>20</th>
<th>50</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td>35</td>
<td>45</td>
<td>25</td>
</tr>
</tbody>
</table>

To confirm that the Excel selection are exported into R correctly, use the *R to Excel* menu, then select *a* or *b* to import back R values into Excel. R commands can now be executed on these vectors, e.g., \( R: \text{mean}(a) \) returns the value 35, and \( R: \text{summary}(lm(a \sim b)) \) summarizes the linear regression of *a* on *b*.

As an alternative, you may also type \( R:a \) or \( R:b \) to read values of vector *a* or *b* from R. Type \( R:a \) then the RETURN key to place the output of the R command in the cell below (as shown in the left picture below). Type \( R:a \) and then the TAB key to place the output of the R command in the right cell (as shown in the second picture below). You can also type \( R:a \) then left-click on any other cell to place the output into the selected cell. **Note:** *if there is any content in the selected cell, the contents will be replaced by the output from R.*

To see the results of \( a + b \) using R, type \( R:a+b \) in any cell, then the RETURN (or TAB) key: this will execute the R command \( a+b \) and place the results back into Excel in the next cell (below the command cell for RETURN or to its right for TAB).

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ra</td>
<td>[1]</td>
<td>10</td>
<td>20</td>
<td>50</td>
</tr>
<tr>
<td>Rb</td>
<td>[1]</td>
<td>15</td>
<td>35</td>
<td>45</td>
</tr>
<tr>
<td>Ra+b</td>
<td>[1]</td>
<td>25</td>
<td>55</td>
<td>95</td>
</tr>
</tbody>
</table>

To produce R outputs in Excel format, use the *Text to Columns* menu under Excel’s *Data* ribbon. The right-hand picture above shows the formatted output results.
For graphics, in addition to the $R$: prefix, CAT uses a $G$: prefix to direct graphics output to the spreadsheet; if $R$: is used instead, then graphics output will appear in a separate window. For example, $G:\text{plot}(a, b)$ produces the following result on the left; $R:\text{plot}(a, b)$ produces the standalone window on the right.

![Diagram of R and G plots]

R functions can also be defined using the $R$: prefix by putting the function definition in one cell. The following picture shows how to define a new R function called $\text{plus}$, and the results of $R:\text{plus}(a,b)$.

**Note:** to enter multiple lines in a single Excel cell, use the ALT-RETURN key combination.

```r
R: plus <- function(a, b) {
  return (a+b)
}
```

```
R: plus(a,b)
[1] 25 55 95 85
```

**Run R scripts**

A block of R commands, each with prefix $R$: or $G$: , can be executed sequentially by highlighting it and clicking on **Run R Script** (at the lower right of the Methods section of the CAT ribbon). The output of R scripts is always placed in the next column to the right of the script being executed, overwriting any data already there.

**Clear R script outputs**

Outputs from R scripts often contain graphics: the normal DEL key only deletes the text. To delete both text and graphs in a selected Excel range, use **Clear Range** (just above **Run R Script**).

**Package utilities**

This utility menu contains shortcuts for the following:

- Load a package from a list of all installed packages
- View all installed packages in a sheet
- Install a new package
- Utilities to install all packages used by CAT
**Function Builder**

CAT provides a unique utility to build an R or CAT function, called *Function Builder*. It helps the user view available R objects (packages, functions, arguments) and avoid typing errors.

To use *Function Builder*, first select a cell as the starting cell to receive R function output. Then click the Function Builder icon (see left picture above). The top line of the function builder will hold the function being built and the arguments (inputs) for the function that have been specified so far. Clicking on the dropdown arrow in the Package line will show all loaded packages. (Click or double-click on a cell in the middle column of the Function Builder window to activate its dropdown arrow.) After a package is selected, click on the dropdown arrow in the Function line to display a list of all functions in the selected package (All user-defined R functions using the R: prefix are automatically displayed in the function dropdown list for package CATkit). After a function is selected, its possible argument values (inputs) are displayed in the first column of the table. (For certain functions, such as CAT_regression, the first argument is called “target”; this corresponds to the dependent variable for the analysis.) To select a single R object as an argument for a function, use the dropdown arrow for the argument. To select multiple objects for a list of arguments, click on the ... button for that line, then swipe to select/deselect multiple of objects (see the right-most picture above). Each selection of the argument line will refresh the top result line. When all arguments have been selected, the top line can be edited if desired, e.g., to change the default R: prefix to G: . Clicking OK executes the function for the specified argument values.

Selecting *View CAT Functions* under the *Function Builder* drop-down menu generates a sheet named *Functions* showing all predefined CAT functions. Any of these can be modified by editing it in this Excel sheet, and then adding an R: prefix to read the revised definition into R. Such customized modifications will be remembered for the Excel workbook. To reset all CAT_functions to their base definitions, use the *Reset to Default* option under the *Build Function* drop-down menu. This will reset the definitions in R, but will not change any edits made in the *Functions* sheet.

*Note: If you’ve made changes to any existing R:CAT_function in a workbook, and reopen the workbook later after upgrading CAT to a new version, remember “Reset to Default” in order to use the updated CAT_functions in the new version of CAT.*
Example: Extending the CAT_describe function

Once the Sample1 data set has been loaded and its columns have put into R using Excel to R, typing the CAT command \texttt{R: CAT\_describe(tmin)} into an Excel cell, or entering via point-and-click it using Function Builder, will produce the results on the following page, including a histogram and a box plot for numeric vectors. Now, suppose that we want CAT to color histograms red. This is not an option in the pre-loaded CAT functions. However, Googling on "R histogram color" will show that a histogram can be colored red by inserting \texttt{col = "red"} into the \texttt{hist()} command. To tie this knowledge about an R command to the CAT\_describe function, select View CAT Functions under the Function Builder drop-down menu. On the resulting Functions sheet, line 3 gives the following base definition of the CAT\_describe(X) function. (Widen row 3 to see the full definition.)

\begin{verbatim}
CAT\_describe <- function(X) {
  print(summary(X))
  grf <- CAT\_beginGraph()
  hist(X, col="red")
  CAT\_endGraph(grf)
  grf <- CAT\_beginGraph()
  boxplot(X)
  CAT\_endGraph(grf)
  library(Hmisc)
  print(describe(X))
}
\end{verbatim}

This code may not mean much to many users who do not use R. Even so, it is easy to use the knowledge that adding \texttt{col = "red"} to the R \texttt{hist()} function colors histograms red to modify the function by replacing the \texttt{hist(X)} line with \texttt{hist(X, col = "red")}. Doing so and prefixing the entire modified function with \texttt{R:} so that it will be processed changes the definition of CAT\_describe for the workbook being used. Rerunning the \texttt{R: CAT\_describe(tmin)} command with this modified function now produces output with a red histogram. This example illustrates how CAT functions can be modified or extended using readily accessible information about underlying R functions.
.Rsession file

After CAT has been installed, any Excel workbook, say Book1.xlsx, will have a corresponding R session file, say Book1.xlsx.Rsession file, which holds all information from the R session that CAT initiates on behalf of the user to perform calculations. This .Rsession file is automatically created and saved by CAT when Book1.xlsx is deactivated, and it is automatically restored when Book1.xlsx is activated. Only the R vectors/factors and data frames created for the active workbook will be active. These will be offered as possible argument values when Function Builder is used.

Note: R functions from all open workbooks are always available and saved in the .Rsession file, regardless of which workbook is active.

One can remove the .Rsession file at any time if it is no longer needed. (The .Rsession file can also be restored from a standard R console if needed using R command restore.session.) Any .Rsession files saved from a standard R console (using R command save.session) can also be also loaded into an Excel workbook if the .Rsession file uses the same name as the Excel workbook: this makes it easy to import R data from any session files into Excel worksheets, then use all the features provided by CAT.

Other Data Utilities

Data utilities described in this section can be used without R to pre-process data before analyzing it.

Recode columns

This utility recodes user-selected columns using a user-defined mapping. Multiple columns can be recoded at the same time. At the top of the recoding window is the default name of the recoded column (an R is appended at the end of the original column header). This default name can be edited. The first column of the table gives the current values for the selected columns in the active sheet, in descending order. To edit the new (recoded) values, double-click the second column, or just start typing in the second column. For the data recoding mapping table, Missing is reserved as a keyword to refer to cells without any value.

Quantile and Decile recoding work differently. They generate new (recoded) columns with values reflecting percentiles for the frequency distribution of values in the original column. Quantile recoding outputs 10, 25, 50, 75, 90, and 100 as the new (recoded) values; a value of 10 means that the original value was in the bottom 10% of all values in that column; a 50 means that it was between the lower quartile and the median value; and a 100 means it was in the top 10% of values. The, Decile recoding is similar, but outputs 10, 20,..., 90, 100 as possible values.
Renumbered recoding will renumber the original values in ascending order consecutively starting from 1.

By default, the new column name for Quantile recoding will have Q added at the end; Decile recoding will have D, and Renumbered will have N added to the end of the original column name. The normal recoding using a user-specified mapping table has R appended to the name of the recoded column. This way, by looking at the tails of the column name, it is easy to see which recoding method was used to produce the column.

The new recoded columns created by recoding are always placed to the right of the table of previously populated columns.

Lags/Delta

This option is used to create a new column from the original by lagging its values by a user-specified number of lags (i.e., shifting all values up by that number of rows). The output may be selected to be either the lagged data, or the delta (difference between lagged and original values) for the given lag. This is useful for time-series and longitudinal analyses. Lagging a column creates rows with missing values at the bottom of the table. Selecting the Clean Rows option under the Split Columns drop-down menu (in the Data section of the CAT ribbon) will remove these incompletely populated rows and should be used so that CAT and R functions that expect columns of equal length will continue to work.

Impute Data

Use this menu to fill in missing data values. When applicable, a new worksheet DataImputed will be created. To use the new data set with its imputed values for missing data, rename the current Data sheet to something else, e.g., DataOriginal, and rename DataImputed sheet to Data, then use the new Data sheet as usual. The imputation method used is predictive mean matching, as implemented in the MICE package.

Split Columns

This option will simply split (recode) the selected column as a set of columns of binary (0-1) values, with a 1 in a specific row and in the column for a specific value indicating that the original column has that value in the same row, and a 0 indicating that it does not. This is useful for converting discrete variables to equivalent 0-1 dummy variables for use in regression or other analyses.

Split sheet dropdown menu will split a data table into several worksheets, using the unique value combinations for the selected columns. This is useful when the data table is large. For example, national data may be split into smaller ones using State and/or county.

Clean Rows dropdown menu simply deletes all rows that contain at least one missing value.
Predefined Menu for User Defined R Functions

There are ten menu slots under the User Defined drop-down menu (currently at the bottom of the Time Series Causality section of the CAT ribbon) that are not pre-populated by CAT. These are for easy execution of user-defined functions. For example, boxplot can be created using Data Explorer for a single column or whole Data sheet. But there is no menu defined to view boxplot for a subset of selected columns. To create a boxplot for selected columns, say PM2.5, AllCause75, tmin, the user could create a one-time R script such as the following:

G:boxplot(PM2.5, AllCause75,tmin)

If this is a frequent task, however, then it is more efficient to create a new user-defined function by entering (or pasting) the following definition into any blank cell:

R:USER_defined <- function(...) {
  grf = CAT_beginGraph()
  boxplot(...)
  CAT_endGraph(grf)
}

Once the function is defined, it will be remembered in the .Rsession file. Now selecting any columns in the Data sheet and clicking on User_defined will generate a box plot for the selected columns.

In addition to the name USER_defined, there are 9 other names reserved for user defined functions, i.e., USER_definedX where X=1 to 9.

Creating new user-defined functions requires knowledge of R, but once they have been created, any user can apply them via the CAT interface by simply selecting columns on the Data sheet and selecting functions to apply to the selected columns by clicking on the user-defined function list.

Decision Analytic Toolkit (DAT) module

This is an initial prototype module. It contains one function, DAT_EU, that calculates the expected utilities for any given decision table. A decision table consists of a rectangular array of numbers with rows representing acts, choices, or decision alternatives that the decision-maker must choose among; columns representing alternative states of nature (possible resolutions of uncertainties or random variables that the decision-maker does not control but that can affect the outcomes of decisions); and values in the cells showing the quantitative outcome or consequence of making each decision if each state occurs. A decision table must be entered on a separate sheet (tab) in a standard format, explained next. Clicking on the Decision Analytics menu icon then causes the output to be displayed in a new worksheet. If the input sheet where the decision table is entered is named DecisionTable, for example, then the output sheet would automatically be named DecisionTable.Output. The Decision Analytics menu can be applied to any worksheet regardless of its name, as long as the inputs on that sheet follow the required format.
A sample decision table is included under menu Excel to R -> DecisionTable, see picture below. The format requirements for providing inputs to the DAT module are as follows:

- **Names of states** are entered in columns, B, C, etc. in row 9
- **Probabilities** of states go in Row 7. These probabilities must be non-negative and sum to 1.
- **Names of alternative decisions** or acts are entered in column A, starting in row 10
- Rows 6 and 8 should be blank. (To avoid invisible whitespaces, these two rows are automatically cleared when running Decision Analytics.)

The drop-down options **Set Probabilities** and **Set Decision Table** under the Decision Analytics menu can be used to set (i.e., import) state probabilities and a decision table from any worksheets. To set probabilities, select a range that contains just one row from any worksheet, then click **Set Probabilities** menu. To set decision tables, select a two-dimensional range that contains both row and column headers from any worksheet, then click **Set Decision Table** menu. The program uses the DecisionTable worksheet to save the values you have set. You may rename this worksheet to any other name to save the scenario. The program will automatically open a new worksheet DecisionTable whenever needed.

There is a built-in default utility function $u(x) = x$, and $\text{DAT\_outcome}(\text{decision}, \text{state})$ which simply returns the value defined in the decision table. To customize the utility function, modify its definition. For example, if you want to use your own utility function, say $u(x) = \log(100 + x)$, then modify it like this:

```
R: DAT_util and R: DAT\_outcome definitions can be stored in any cell of any worksheet. Names daProb and daTable are reserved R names by CAT in R. You can see their R values by typing R: daProb or R: daTable in any cell.
```
Predictive Analytics Toolkit (PAT) module

The CAT functions discussed so far are useful for interactively exploring and visualizing relations among the variables in a data set. Sometimes, however, it is desirable to automatically fit and evaluate many different predictive models. The Predictive Analytics Toolkit (PAT) module allows for fully automated development and testing of predictive models, using the framework of the CARET package in R. To do so, one can simply use the Predictive Analytics ribbon. For large data sets, however, filters can be used to speed up the modeling. If desired, additional models can be selected from the All Models tab; a selection of the most useful and frequently used predictive models are shown under the Models tab.

Note: The PAT module is not included in the free version of CAT. For more info, contact tcoxdenver@aol.com

The General tab for the PAT module allows the user to specify the following choices:

1. *The percentage of the data set (from 10% to 100%) to be used in training the predictive models.* Once the user selects a percentage using the slider control, PAT will create a data partition that allows part of the data set (the percentage specified by the user) to be used for fitting and optimizing predictive models. The rest of the data records – those that have not been used to build the models – are then used to evaluate the predictive performance of each model on this “out-of-sample” data. (Statisticians and machine learning experts should note that efficient data use techniques such as cross-validation and bootstrapping can be used within the training set, but a disjoint test set is still reserved to allow an objective assessment of out-of-sample performance, unless the user sets the “Train data percentage” slider to 100%. In that case, the same data are used to both train and test models, and overfitting is avoided, if at all, only by using each predictive algorithm’s own resampling techniques.)

2. *Whether to create a balanced, randomized data partition* or to instead use the initial records at top rows of Data sheet to train and the rest to test. If “Use top rows as train data only” is not checked, then PAT follows that CARET methodology (download documentation pdf here www.jstatsoft.org/article/view/v028i05) in creating stratified random splits of the data (using createDataPartition in CARET). If it is checked, then the top records in the data set are used to
train and the rest to test; this can be useful in forecasting applications, where the data that becomes available first is used to forecast patterns in subsequent cases.

3. **Use all data to predict NA values.** When checked, all available data in target will be used to predict NA values in target. Since no data is used as test class data, performances of each model will not be tested.

4. **Select filters.** If there are thousands of columns (potential predictors) in the data set, then those that are identified as the most significant predictors of the dependent variable in CART trees, linear regression models, or randomForest importance analyses can be automatically selected by checking the appropriate filter. If multiple filters are selected, then PAT will keep variables that pass each of them.

5. **Pre-process data.** Redundant variables (those that are highly correlated with each other) can be removed and the remaining variables standardized and filtered to remove low-variance predictors by checking the “Remove highly correlated” and “Preprocess predictors” options, respectively. (See www.jstatsoft.org/article/view/v028i05 for details.)

For small data sets with only a few dozen columns, filters and pre-processing options can be left unchecked. For large data sets with thousands of columns, they can substantially reduce run times. Users should expect run times of several minutes even for small to moderate sized data sets.

The **Models** tab allows the user to select specific popular predictive models to include in the PAT analysis. For most purposes, this tab can be ignored and PAT will then use a default set of models including randomForest, ctree and rpart (two different CART tree algorithms), and earth (the R renaming of Multiple Adaptive Regression Splines (MARS), which is trademarked and licensed to Salford Systems). Gradient boosting machines (gbm) and boosted generalized linear model (glmboost).

The **All Models** tab provides access to a comprehensive library of predictive analytics models. These can be checked to include them in a PAT run.

**Caution:** Some of the algorithms in the All Models library may be very slow, and not all of them are guaranteed to work, as support for the various packages is variable, some are out of date, and maintenance of this library is not provided by the CAT/PAT development team. The models under the Models tab are supported (although even some of those may not converge for particular data sets, leading to empty outputs for those models), and should provide a useful and robust black-box predictive analytics capability for most practical applications.

The **Train Control** tab is for use by experts in machine learning who are familiar with the CARET framework and algorithms. It allows the parameters of model training routines to be customized as desired. For most purposes, this tab should not be used, as the default values and built-in parameter optimization using grid search will suffice to automatically generate approximately the best predictive models that can be found for the data provided.

Once the PAT module has been initialized by selecting models to include (or accepting the recommended defaults), the selected models are all fit to the data selected by the user. The first variable (column) selected by the user of the Data sheet should be a column of 0 and 1 values: this is the dependent variable to be predicted from the remaining columns that are selected. Loading Sample3
under the *Excel to R* icon at the far left of the CAT ribbon and selecting columns A-L, then clicking on the Predictive Analytics icon in the “Predict” section of the ribbon generates the following outputs. To follow the details of the outputs, this sample data set should be used. It is a data set showing which chemicals are classified as mutagens, and the goal of predictive analytics is to generate models that accurately predict whether each chemical is classified as a mutagen (value of 1 for the “mutagen” variable) or is not a mutagen (value of 0 for the “mutagen” variable). The full data set from which this sample was taken is described here: [https://cran.r-project.org/web/packages/QSARdata/QSARdata.pdf](https://cran.r-project.org/web/packages/QSARdata/QSARdata.pdf).

The outputs of a PAT run are as follows. These outputs are collected in a PAT-generated tab named Predict.

First, a graphical summary of the confusion matrix is provided for each predictive algorithm. In the following example, earth, rpart, ctree, and rf (randomForest) were the selected algorithms. The green lobes indicate correct predictions and the yellow ones indicate errors. For the earth algorithm, 39 cases were correctly predicted to be 0 (not a mutagen) and 80 were correctly predicted to be mutagens. There were 43 false positive and 12 false negative predictions.

Next, a table is generated summarizing the performance of each algorithm on each of several standard metrics such as accuracy, sensitivity, and specificity; see [www.dataschool.io/simple-guide-to-confusion-matrix-terminology/](http://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/) for definitions of these terms.
Areas under the ROC curves for different models are also summarized and displayed, with values near 1 indicating nearly perfect classification and values near 0.5 indicating no better than random classification.

PAT includes a diagram not found in other packages that clusters records (e.g., chemicals, in this case) and that shows how the values predicted by each algorithm (red = 0, yellow = 1) compare to the observed values (top row). Chemicals for which a vertical column is all red or all yellow are predicted correctly by all methods, suggesting that they are relatively easy to classify. Columns with mixed white and red indicate cases that are more difficult to classify. PAT also tabulates these data on what was observed and what each method predicted, to the right of the other outputs.
Finally, PAT generates several other outputs, including calibration curves that indicate the fraction of cases with different predicted probabilities of being classified as 1 (x axis) that actually were classified as 1 (y axis). The 45 degree line indicates perfect calibration. The following calibration curves show that rpart gives relatively well-calibrated predictions (class probabilities, which are then rounded up to 1 or down to 0 to get a final predicted classification) in this example, but ctree gives poorly calibrated predictions, with far more than the predicted number of chemicals being classified as mutagens among those that ctree predicted as having low probabilities of being so classified; and far fewer than predicted being classified as mutagens among those for which it predicted high probabilities of being classified as mutagens. This poor calibration explains the relatively large yellow lobes in the initial visualization (above) of the confusion matrices for these classifiers.
**Conclusion**

This concludes our summary of CAT’s and PAT’s core capabilities, together with the initial phases of the DAT module. For most users, the main purpose of CAT/PAT is to give simplified access to the analytics power of the vast array of R packages and commands that are useful for detecting, analyzing, quantifying, and visualizing associations and other relations (such as information relations among multiple variables) in data sets using standardized, well-documented, and well-supported algorithms. For advanced users, CAT provides a convenient way to integrate R programming directly into Excel, while also providing pre-built commands with simplified syntax, push-button analytics capabilities that can save time on routine tasks, and reports that often integrate the analyses from multiple R packages to provide different perspectives on relations in the data.

We encourage users at all levels to use CAT in conjunction with Google. Google can be used to find useful R packages and capabilities and CAT can then be used to install additional packages as needed and to access them via CAT’s simplified interfaces. More capabilities are likely to be added over time, and CAT will be updated as R releases and packages are updated and new packages for advanced analytics and causal analysis are added to the CRAN repository. We therefore encourage users to check for CAT updates often [here](mailto:tcoxdenver@aol.com). We also welcome and encourage users to send comments, questions, notifications of bugs or difficulties in using CAT/PAT, and suggestions for improvements and additions to tcoxdenver@aol.com.

We hope to make CAT/PAT and its add-on modules as useful and easy-to-use as possible. Your feedback will help to achieve this goal.