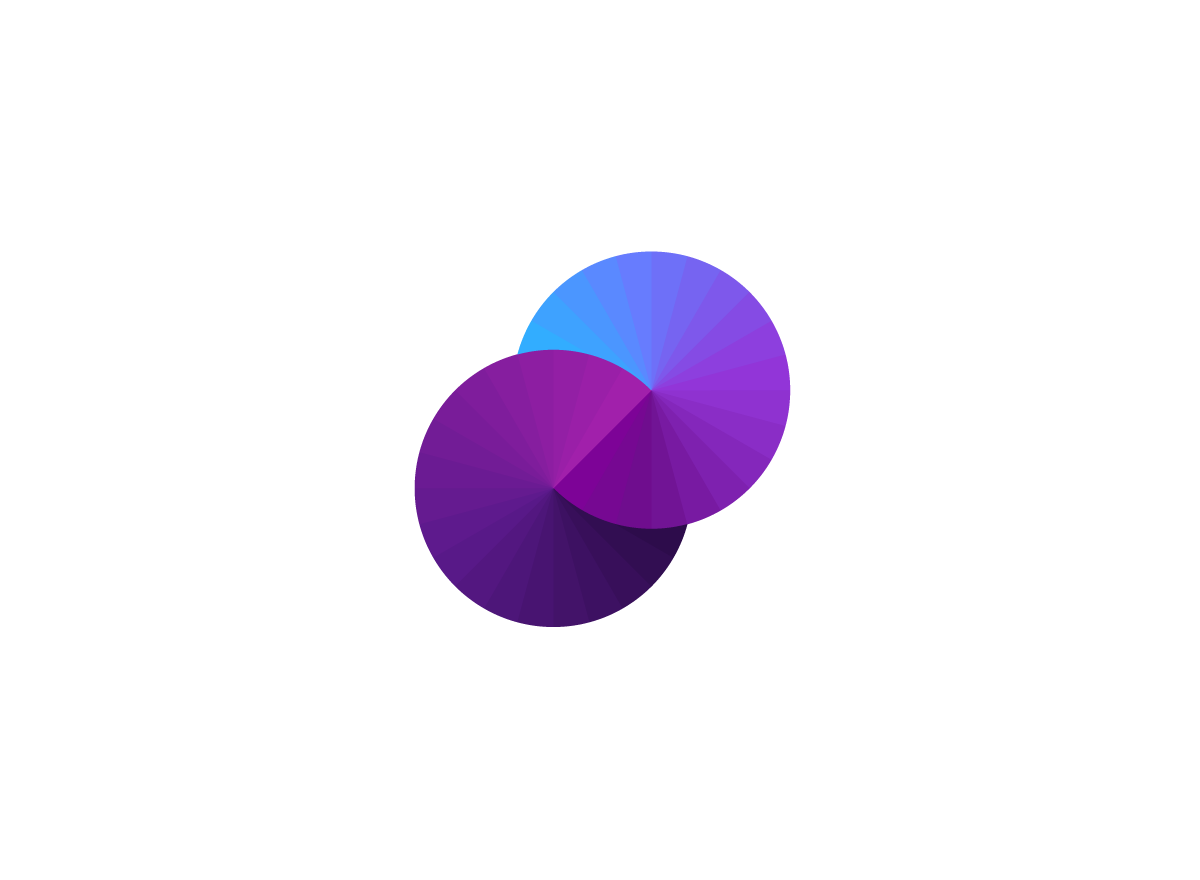
Getting started with LMMM package



By Jason Dealey ([Jason.dealey@mindshareworld.com](mailto:Jason.dealey@mindshareworld.com))

Mindshare UK Office

Version 2.4.0

Contents

[Setup 4](#_Toc515619975)

[Example 1: Using lmmm’s built in UI (single equation) for a linear model 5](#_Toc515619976)

[Naming of data variables 5](#_Toc515619977)

[Launch the shiny app and import the data 6](#_Toc515619978)

[Through RStudio 6](#_Toc515619979)

[Through the shortcut 7](#_Toc515619980)

[Using the built in UI 7](#_Toc515619981)

[Initial Setup 8](#_Toc515619982)

[Explore Data 12](#_Toc515619983)

[Multi Variable 12](#_Toc515619984)

[Correlation Heatmap 12](#_Toc515619985)

[Panel Data 13](#_Toc515619986)

[Variables 14](#_Toc515619987)

[Manage Variables 14](#_Toc515619988)

[Create Variables 14](#_Toc515619989)

[Random Period Dummy and Multi Period Dummy 15](#_Toc515619990)

[Diminish Decay Single and Multi (adstock) 15](#_Toc515619991)

[Lag/Lead 16](#_Toc515619992)

[Transform (exp, log etc.) 16](#_Toc515619993)

[Multiply/Add variables together 16](#_Toc515619994)

[Panel Level – Divide by cross sectional mean 16](#_Toc515619995)

[Insert Excel Time Series 16](#_Toc515619996)

[STL (Seasonal and Trend decomposition using Loess) 17](#_Toc515619997)

[Hodrick-Prescott filter 17](#_Toc515619998)

[Manual Code Entry 17](#_Toc515619999)

[Model 18](#_Toc515620000)

[The left hand side 18](#_Toc515620001)

[The right hand side 19](#_Toc515620002)

[Model 19](#_Toc515620003)

[Time Window 21](#_Toc515620004)

[Diagnostics 23](#_Toc515620005)

[Contribution 24](#_Toc515620006)

[Test Variables 25](#_Toc515620007)

[Test Diminish 26](#_Toc515620008)

[ROI 27](#_Toc515620009)

[Genetic Algorithm 28](#_Toc515620010)

[Choose Vars 28](#_Toc515620011)

[Output 29](#_Toc515620012)

[Set Defaults 30](#_Toc515620013)

[Example 2: Using lmmm’s built in UI (single equation) for a panel level or mixed effects model 31](#_Toc515620014)

[Initial Setup 31](#_Toc515620015)

[Explore 33](#_Toc515620016)

[Panel Data 33](#_Toc515620017)

[Var Create 33](#_Toc515620018)

[Panel Level – Divide by cross sectional mean 33](#_Toc515620019)

[Model 34](#_Toc515620020)

[Appendix A – Changing reference points 36](#_Toc515620021)

[Appendix B - Adstocking 37](#_Toc515620022)

[Appendix C – Log Decomposition explanation 38](#_Toc515620023)

[Appendix D – Dealing with Collinearity 39](#_Toc515620024)

[Appendix E - Notes on Mixed Effects 40](#_Toc515620025)

[fixed and random effects with REML 40](#_Toc515620026)

[The Optimiser, and “failures to converge” 40](#_Toc515620027)

[Should we even be running a mixed effects model? 40](#_Toc515620028)

[Other links 41](#_Toc515620029)

This Getting Started Guide will take you through how to set up and use the lmmm package for R.

lmmm stands for Linear Marketing Mix Model. It is built around an R Shiny tool to combine the Mindshare market mix modelling workflow as an easy to use point and click interface.

# Setup

There are several programs that must be installed the first time the program is run.

1. **R**

The lmmm package is built in R. When installing R, it is recommended to install it directly to the C drive rather than Program Files. Some packages don’t like the space in ‘Program Files’. The recommended location is therefore C:\R\R-3.5.0 (at time of publication this was the latest version). Make note whether R asks you to install just a 32bit version or both a 32bit and 64bit version.

1. **RStudio**

RStudio is a set of integrated tools designed to help you be more productive with R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management. Working with R through RStudio is preferred by almost all R users.

1. **Java Offline**

Java Offline is required for exporting certain outputs from the shiny console. If in step 1 you were asked to install both a 32bit and 64bit version of R, install the 64bit version of Java Offline. Otherwise, install the 32bit version of Java Offline

<http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>

1. **Install packages required for lmmm**

To run the package you need to install all package dependencies (if they are not already installed). To do this open RStudio. One of the four separate panes should be called console. Copy and paste the below code into the console and press enter.

install.packages(c('plyr', 'dplyr', 'googleVis', 'shiny', 'lubridate', 'data.table', 'lme4', 'DT', 'scatterD3', 'XLConnect', 'reshape2', 'HH', 'lmtest', 'tseries','rhandsontable','pbapply','jsonlite','devtools','plotly',

'rgenoud','parallel','doSNOW','tidyr','openxlsx','shinyjs','optimx','RSQLite'))

library(devtools)

devtools::install\_github("jrowen/rhandsontable")

If you have trouble installing these packages, you can select them manually from the packages menu on the RHS (the same section you find help and plots in)

1. **Install the lmmm package**

Then you will need to install the lmmm package itself from the zipped file. Similar to step 4, copy and paste the following code in the console, change the location in the code to where you have saved your zip file (Note: you need a double backslash or a single forward slash in the location for this to work properly. Also, this should link to the entire zip folder).

install.packages("C:\\Users\\YourName\\Downloads\\lmmm\_2.4.0.zip", repos = NULL)

Common errors in running this line of code might relate to there being a space in the location of the R installation (e.g. “C:/Program Files/R/R-3.5.0”). In that case, revert to step 1 and reinstall R to a location without spaces (e.g. “C:/R/R-3.5.0”).

1. **(Optional) Copy the link to run directly from your desktop**

Go to the location that lmmm has been unpacked. This should be C:\R\R-3.5.0\library\lmmm\Run if you followed the above save locations, otherwise it will be wherever you saved R then \lmmm\Run.

Edit the LMMM.bat file in notepad and change C:\r\r-3.5.0\bin\Rscript.exe to the location of Rscript.exe file in your directory. If everything has been followed, nothing should need to be changed. If you are not updating R, to find the location of R, go to Tools>Global Options in RStudio.

Copy the LMMM shortcut to your desktop.

Test by double clicking on the LMMM shortcut on your desktop. This should boot up a command prompt as well as open the tool in your default browser.

If you are installing from either **Linux** or **MacOS**, please contact us to get hold of a .tar.gz version and instructions on installing.

# Example 1: Using lmmm’s built in UI (single equation) for a linear model

As an example we will walk through a piece of analysis of some magazine sales for a celebrity magazine. This data comes with the package, or will be attached as a csv. First, we’ll talk about naming variables

## Naming of data variables

Here we assume the data is already prepared and ready for analysis (to see which functions can assist with the data prep task/stage, please refer to the dataprocess package). HOWEVER, it is important that data is appropriately labelled.

The variables get grouped based on the first character in their name. A list of these can be found below. It is possible to manually change these later, but adhering to this list makes the modelling process easier.

This helps in situations such as testing or viewing all variables from a certain group, scrolling through the variable list, running a log-lin model where the way variables are grouped varies the output contributions and return on investments, and in the genetic algorithm to save modelling time.

|  |  |  |
| --- | --- | --- |
| **Expression** | **Group** | **Example** |
| Full name = intercept | BASE | Intercept |
| Begins with b or B | Base | B\_trend |
| Begins with c or C | Competitor | C\_CompetitorMedia |
| Begins with d or D | Distribution | D\_ownDistribution |
| Begins with e or E | Economy | E\_ConsumerConfidence |
| Begins with m or M | Media | M\_tv\_grp\_campaign1 |
| Begins with o or O | Offers | O\_30percent\_Off |
| Begins with p or P | Price | P\_ownPrice |
| Begins with r or R | PR | R\_PositiveMentions |
| Begins with s or S | Seasonality | S\_JanuaryDummy |
| Begins with u or U | Unknown | U\_dummy01Jun16 |
| Begins with v or V | Events | V\_Olympics2012 |
| Begins with w or W | Weather | W\_HoursOfSun |
| Begins with y or Y | OwnVar | Y\_volume\_sales |
| Any others | Unknown | aRandomVariable |

For media variables, a similar format should be in existence for mapping variables to their spend variables (subject to them existing as a column in the dataset). Both the media variable and spend variable take the form m\_X\_grp\_Y and m\_X\_spend\_Y and where X and Y relate to the individual. recommended to use X as the channel, and Y as more information about the campaign (Note: no underscores or full stops are allowed). This gets used when calculating ROIs and media curves within the tool.

## Launch the shiny app and import the data

### Through RStudio

 **Task:**

Open RStudio. You should begin by loading the packge into memory in R.

library(lmmm)

To know that this has run properly, the following bit of text should be displayed at the bottom of the console

Attaching package: ‘lmmm’

The following object is masked from ‘package:data.table’:

Shift

…….

There are two ways to get data into the modelling platform. The first is directly in R (detailed below); the second is via a comma-separated-values (csv) file directly in the tool (detailed further down)

R can import any format of data, a google search for ‘R import excel’ for example will detail how to import data from excel. However, in this example, we will use the data “celeb.mod.trans”. This is saved within the package so you just need to load it up.

data(celeb.mod.trans)

Once loaded, you will see celeb.mod.trans on the upper right hand side of the page under environment. Clicking on this opens the data to view. An alternative way of bringing this up is by running

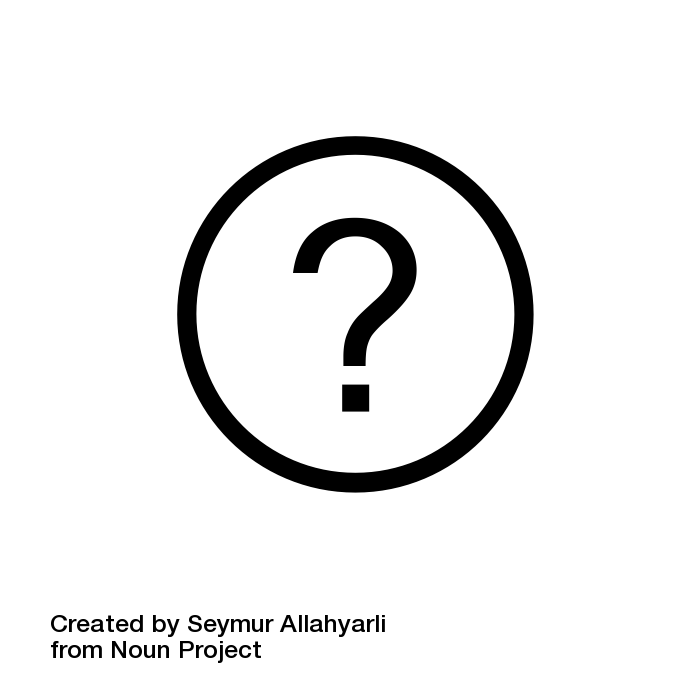
View(celeb.mod.trans)

Run the buildModel tool

buildModel(celeb.mod.trans)

To stop running the buildModel function, you should go into Rstudio and click on the stop button on the console page, in red below. Without doing this, any additional statements entered in R won’t run.



 **Tip:**

There are a number of additional arguments to the buildModel function. Bringing up the help file with ?buildModel in the console will open a help file. Every R function has a help file that you can access using the ?. Typing apropos(“model”) will show all functions that contain the word model in them.

### Through the shortcut

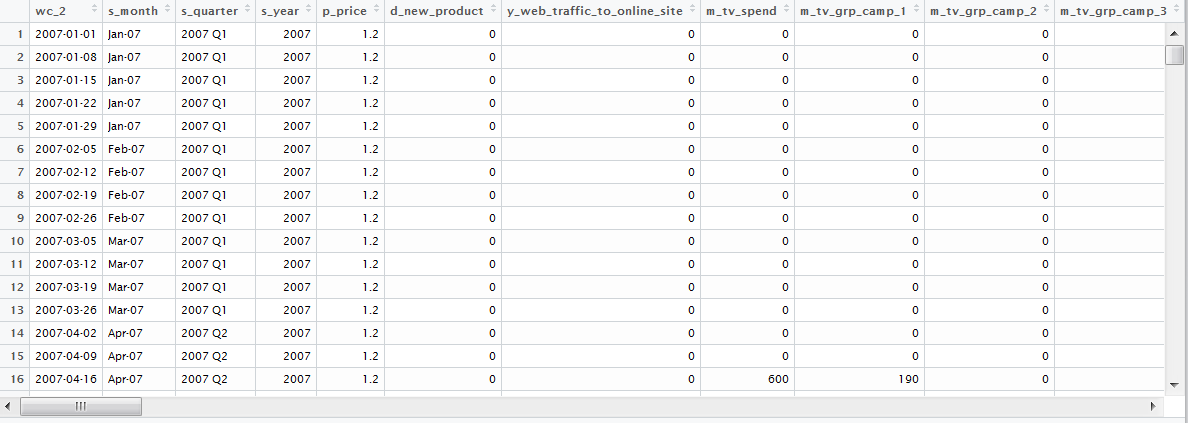
 **Task:**

Double click on the shortcut icon on your desktop. This should launch a command prompt which runs R, and then launches the modelling tool in your default browser.

Upload the csv dataset called celeb\_mod\_trans.csv that came attached with this package under the ‘Load Dataset’ button

The data is in a wide file format. There is:

* A date variable – wc\_2
* The dependent variable – y\_sales
* Independent variables



## Using the built in UI

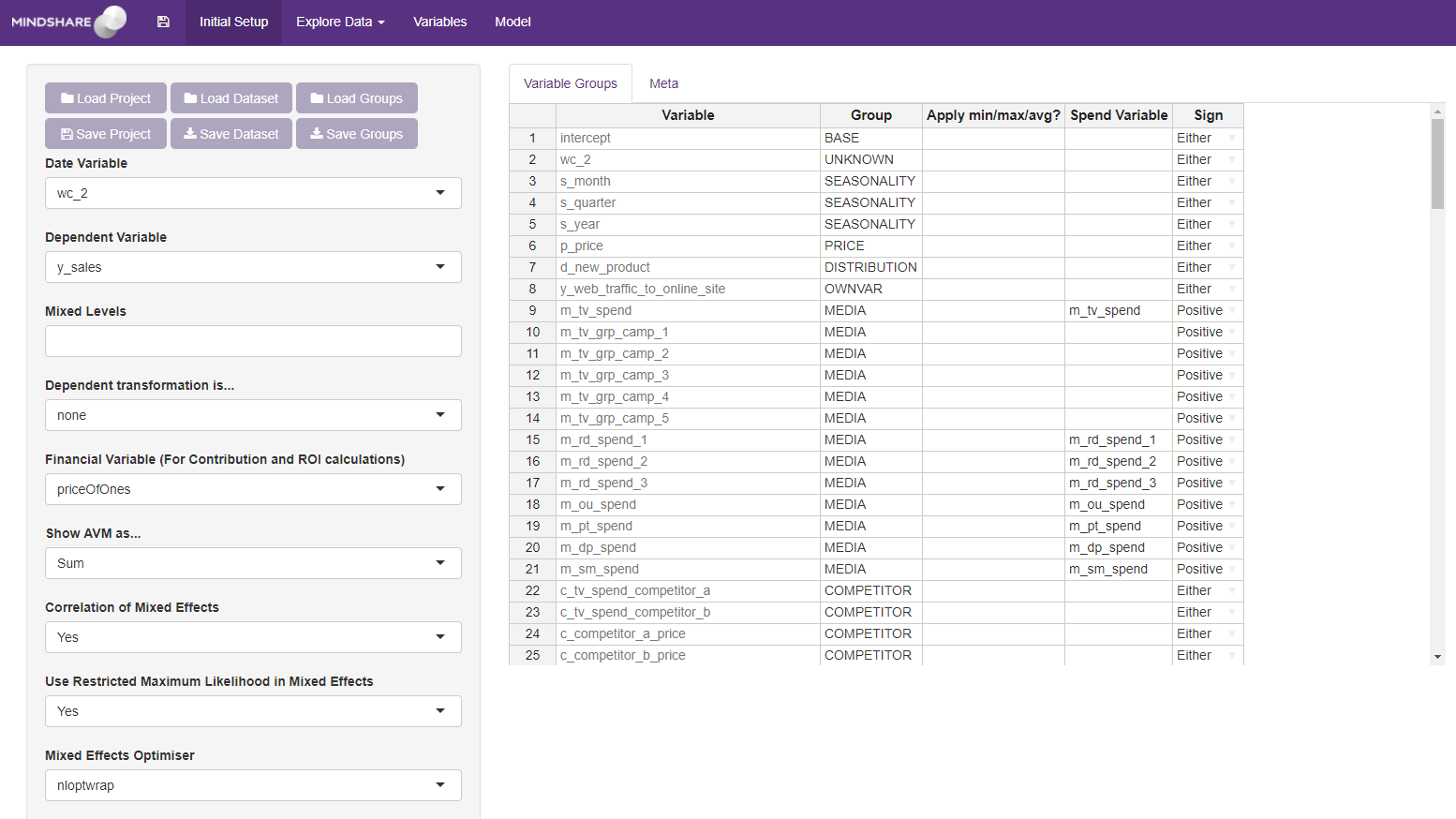
Running either task above will open a new tab in your default web browser. Within this the user can save, or navigate to four (sequential) main tabs, each performing a different task:



* A save button – Saves a .lmmm file to the download bar (May need to unblock popups)
* Initial Setup – Define key variables such as date variable, dependent variable and transformation as well as defining variable groups
* Explore Data – Three separate pages for viewing data in common charts
* Variables – Create and manage variables on the fly, such as adstocks, lags, and dummys
* Model – Run models and look at the outputs from fit and diagnostics through to contributions and ROI.

## Initial Setup

Inputs that get defined in the initial setup have ramifications throughout the rest of the dashboard and should therefore be considered to the highest degree before stepping through the rest of the tabs.



On the ‘Initial Setup’ tab, several fields are required to be completed by the user:

1. **Load and Save Project**

Save project acts identically to save button in the navigation bar. This saves out a .lmmm file which can be loaded at a later date. It contains everything to pick up where you left off, including the data, all variables created, saved models, and variables in the model.

This file can be loaded directly into the tool using the Load Project button.

1. **Load and Save Dataset**

The Load and save dataset allows a user to upload/download a csv of the data in the current project. It currently only accepts csv.

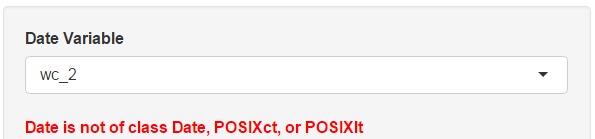
 **Task:**

Upload the dataset celeb\_mod\_trans.csv to the application

1. **Date Variable**

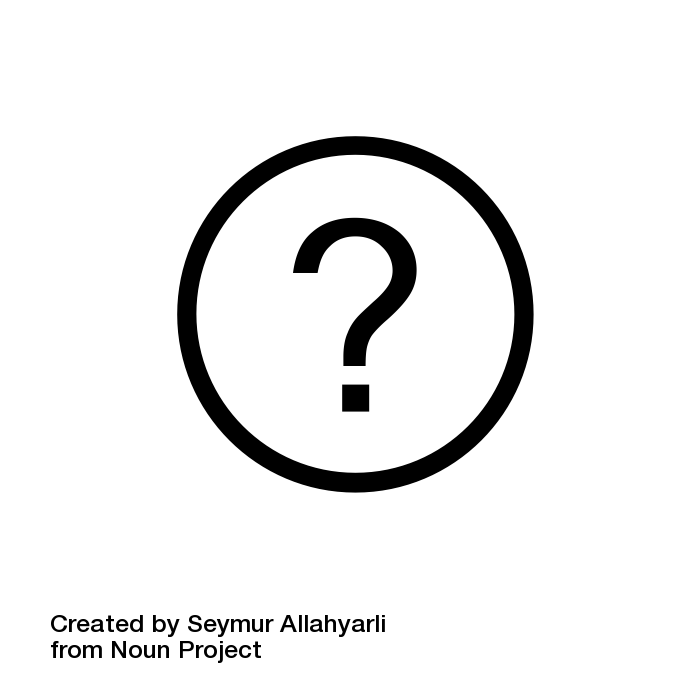
The date variable in the dataset. Set this to be ***wc\_2***.

The date automatically tries to convert the selected column into a date variable. If it encounters errors (i.e. returns NAs) then the following error will occur.



 **Task:**

Set the date variable to be wc\_2

 **Tip:**

The date variable can be set using the argument date.var in buildModel

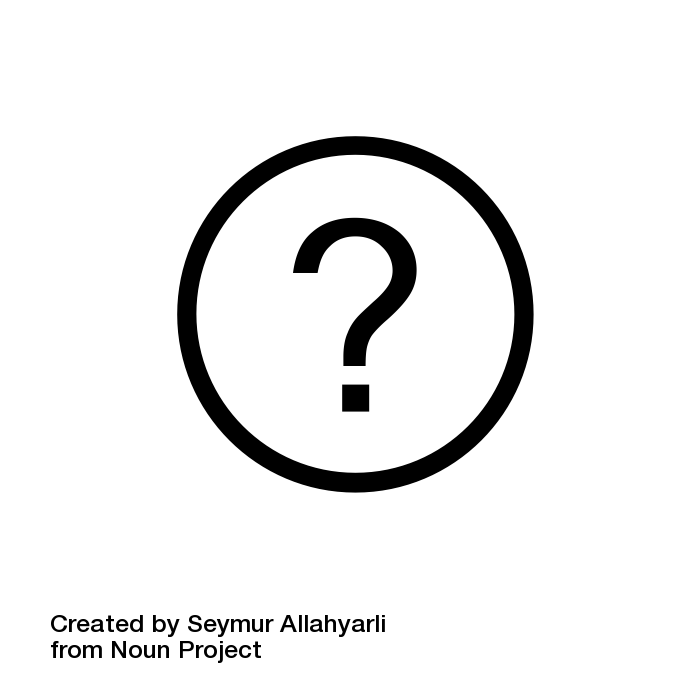
To get the correct **format** for as.Date, type ?strptime in the console

1. **Dependent Variable**

The variable to be modelled in an untransformed format.

 **Task:**

Set the dependent variable to be y\_sales

 **Tip:**

The dependent variable can be defined in the formula argument of buildModel. The correct way to write this would be dependentVariable ~ IndependentVariable1 + IndependentVariable2. In future runs, setting these make the setup a lot easier and repeatable.

1. **Mixed Levels**

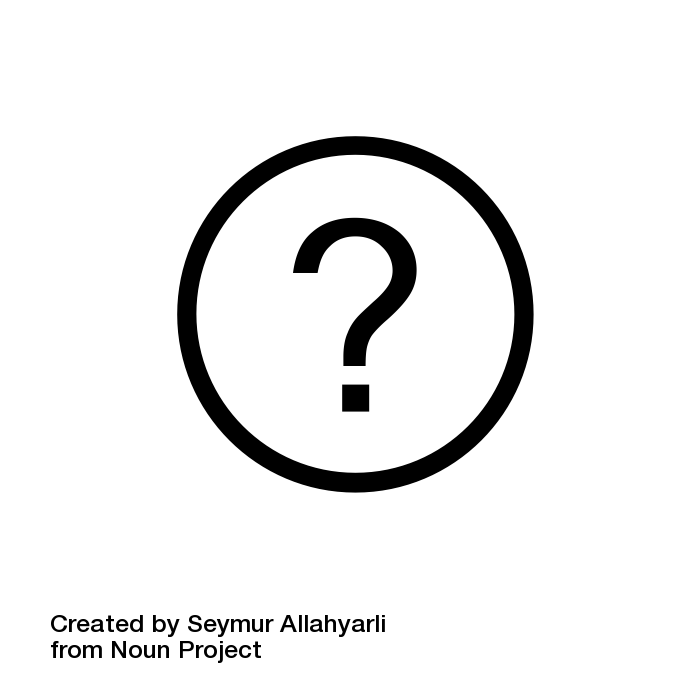
Not applicable in a linear model. See [example 2](#_Example_3:_Using) for an explanation. Leave this blank.

1. **Dependent transformation**

Defaults to ‘none’. Can change to either ‘natural log’ or ‘mean divided’ if you want to transform your dependent variable in either of these two ways.

 **Task:**

Set the dependent variable transformation to None.

 **Tip:**

The transformation can be set by the argument depVarType in buildModel.

Set this to none to keep the dependent variable as it is;

Set this to ‘natural log’ to apply to natural log to the dependent variable;

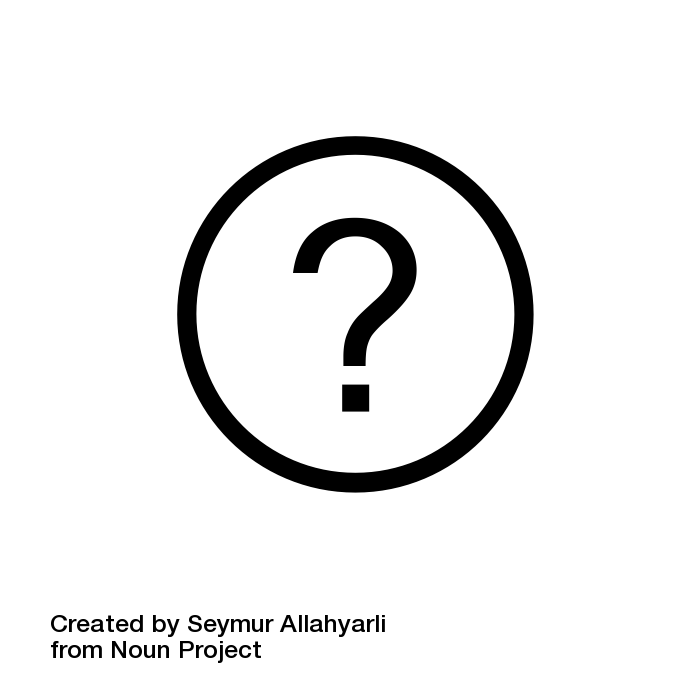
Set this to ‘mean divided’ to divide by the mean sales of the dependent so the variable varies around 1 (More explanation in Example 2).

1. **Financial Variables**

The financial variable should be a variable in the dataset that turns the metric you are modelling into a financial calculation. It could turn volume into value by using price per unit, or could turn volume into margin by using profit per unit. Set this to ***p\_price***.

 **Task:**

Set the price to p\_price

 **Tip:**

The price variable can be set by the argument price in buildModel.

If you do not require a financial variable, this can default to a variable called priceOfOnes, which is a dummy variable of ones that gets created

1. **Show AVM as …**

Options of “Sum” or “Average” for viewing the Actual vs Model on the Model tab.

1. **Correlation of Mixed Effects**

Not applicable in a linear model. See [example 2](#_Example_3:_Using) for an explanation. Leave this default.

1. **Restricted Maximum Likelihood**

Not applicable in a linear model. See [example 2](#_Example_3:_Using) for an explanation. Leave this default.

1. **Mixed Effects Optimiser**

Not applicable in a linear model. See [example 2](#_Example_3:_Using) for an explanation. Leave this default.

1. **VariableGroups**

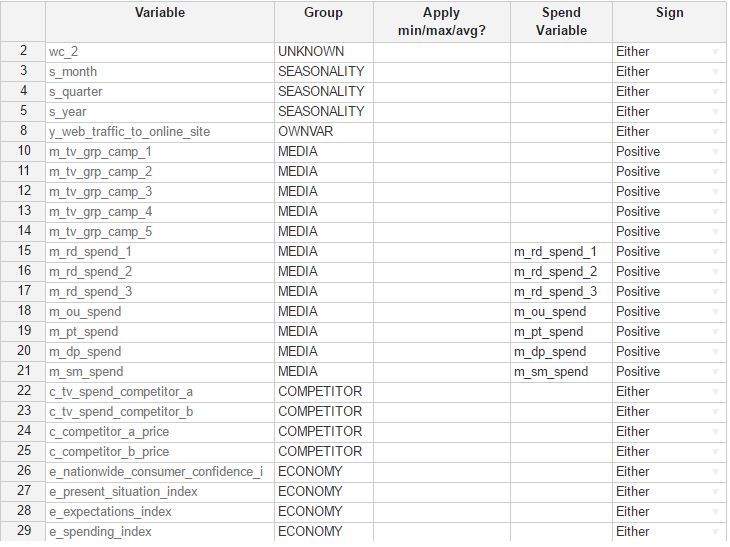
These variableGroups are editable, and new ones can be created. Similar to excel, it is possible to change multiple at the same time by selecting them changing the name and then pressing ctrl+enter. The groups are most used when running log modelling where variables of the same group get put together – more comments on this in the modelling section.

Apply min/max/avg? refers to adjustments that should be made the original variable when calculating contributions. This has large ramifications for log modelling[[1]](#footnote-1), however just shifts values into the base for linear modelling, changing the interpretation of the variable.

SpendVar automatically links any media variable to its corresponding spend variable if it exists in the dataset and they both take the form m\_X\_spend\_Y and m\_X\_grp\_Y where X and Y are the variable name (Note: no underscores or full stops are allowed). This gets used when calculating value contributions and also in the ROI tab.

Sign helps you to spot variables in your model that have a sign opposite to what you expected. Options for this column are Positive, Negative, Either, or Exclude. Exclude is mostly for the genetic algorithm which we go through later on in this help document.

To download this table, you can either click on the download tab on this setup page, or click save Instance on the model page.

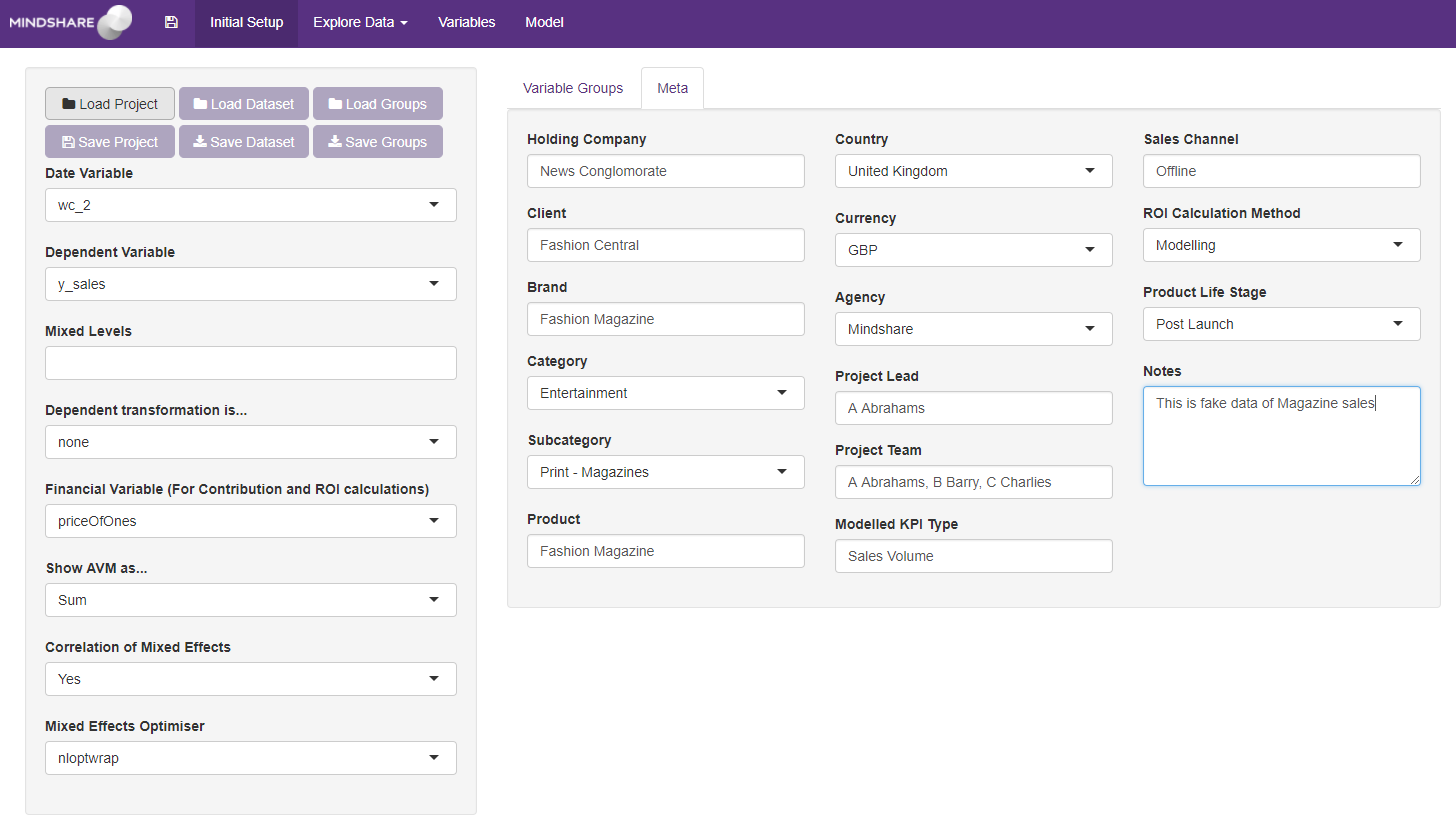


 **Task:**

The variables e\_oscarXXX have been given the wrong starting character. Change these in the tool to be Events. Using ctrl+enter on multiple cells will change them all at the same time.

1. **Meta**

The Meta tab must be filled in when uploading data to the ROIDatabase (see [ROI](#_ROI)). Cells that have to be filled are highlighted in red at the top.

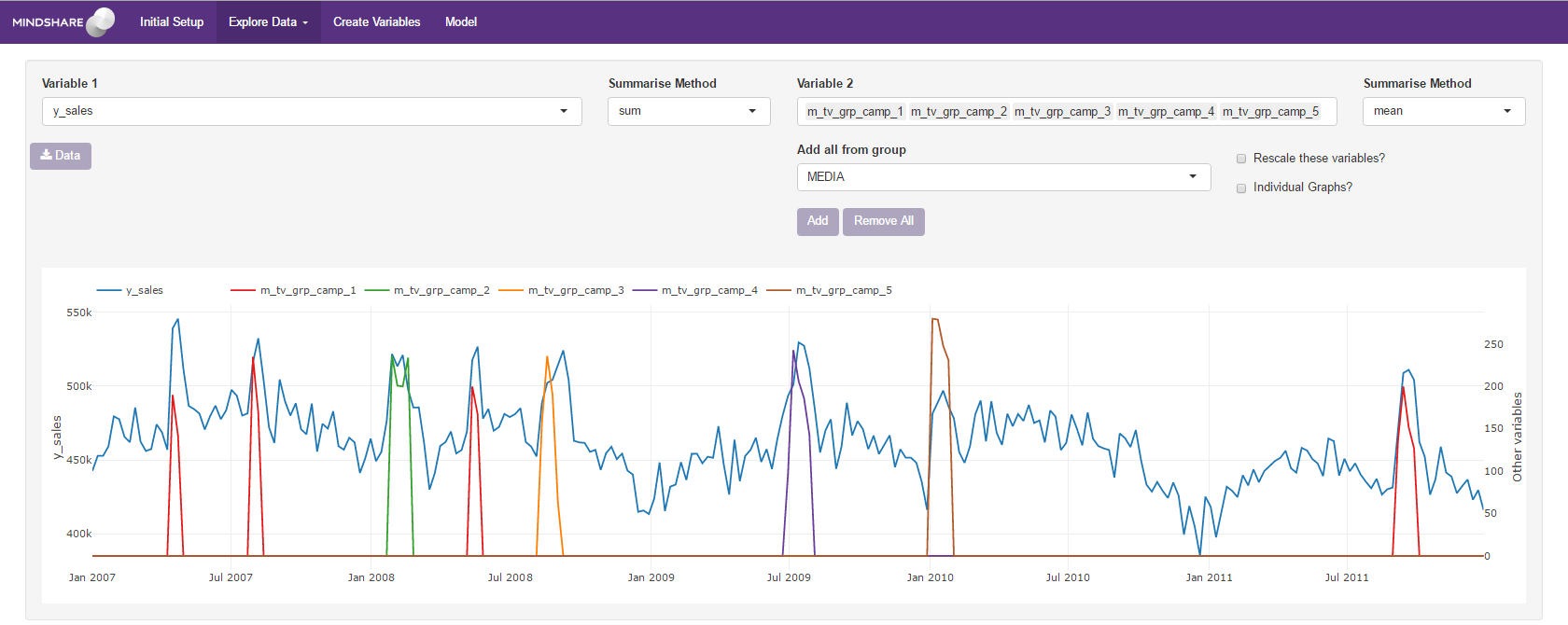


## Explore Data

There are currently 3 views on the explore data tab

* Multi Variable – plot multiple variables either on the same graph or separate
* Correlation Heatmap – Show the top correlations for one variable, or a correlation matrix for all variables selected
* Panel Data – Explore two variables across two different regions in a scatter and line chart

### Multi Variable

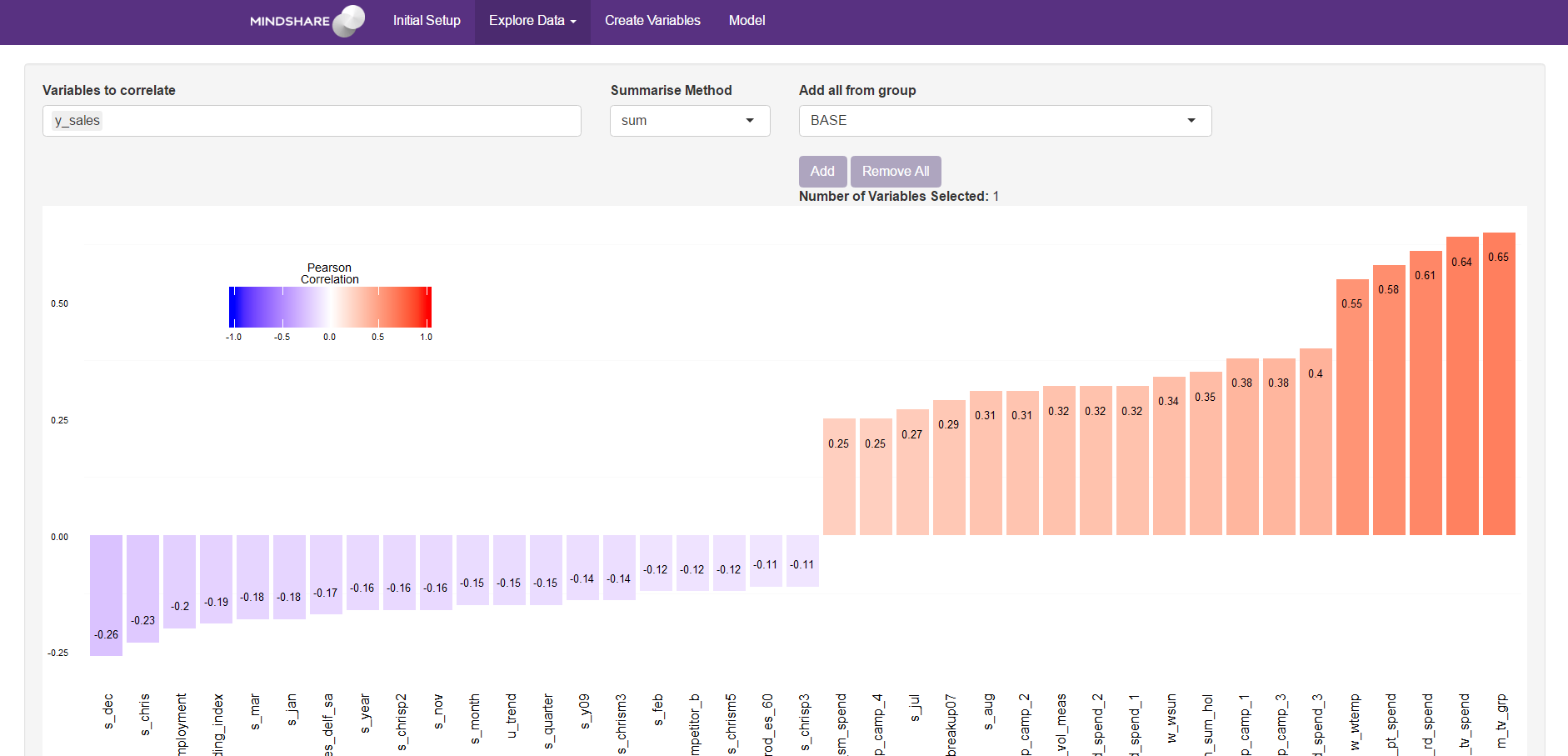


The multi variable tab allows national level plots of numerous variables at the same time on the right hand axis against one variable of interest of the left hand axis. It is possible to add all variables from a particular group (see variableGroups) and it is also possible to view each of the Variable 2 variables on a separate graph by clicking the “Individual Graphs?” button.

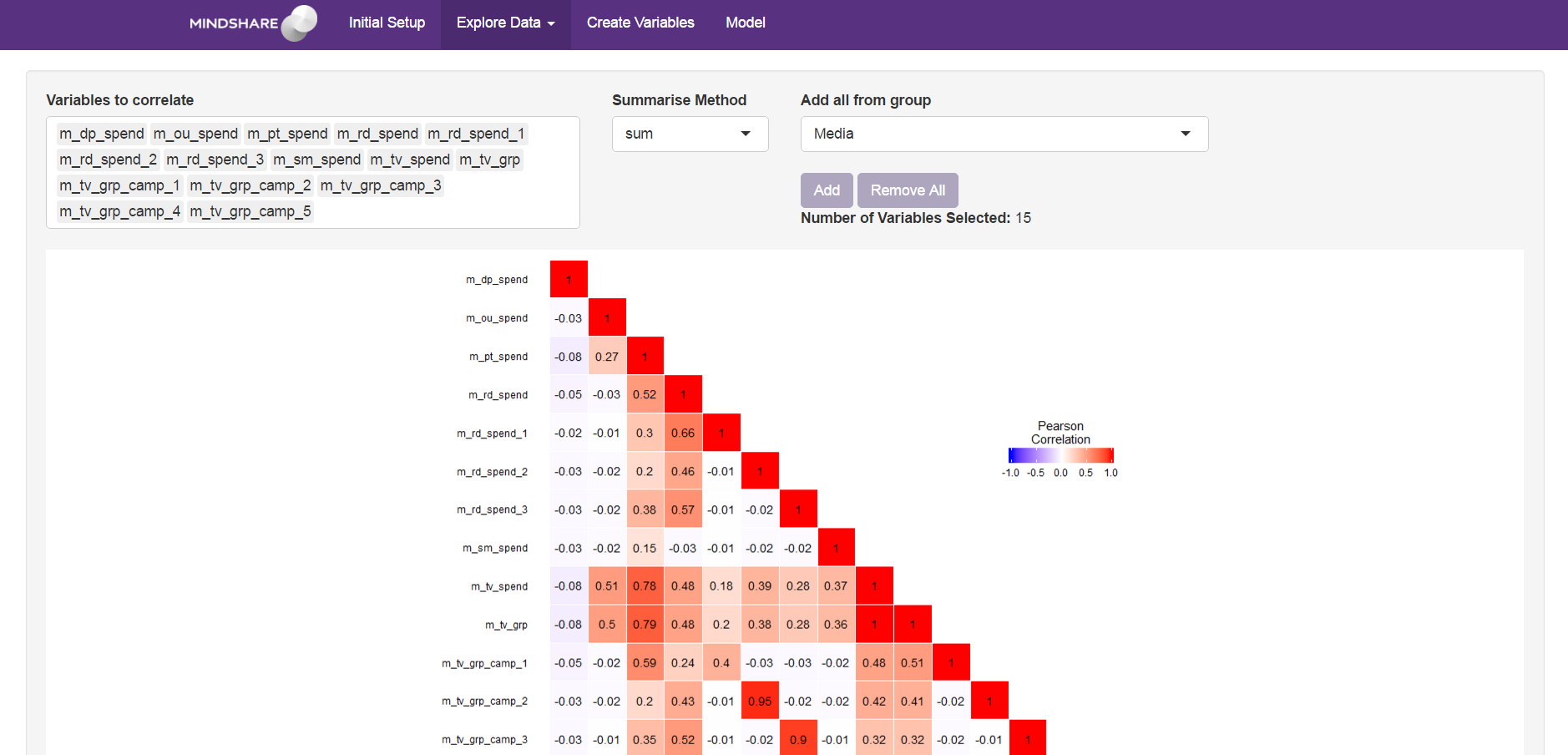
 **Task:**

Add the 5 TV campaign variables (m\_tv\_grp\_camp\_X). Study the graph. Select Media from the add\_all selection and click add. Notice the multicollinearity between some of the variables

### Correlation Heatmap



When a single variable is chosen, the correlation heatmap displays the highest and lowest 15 correlations

When multiple variables are chosen, the correlation heat map displays the correlation between all of the variables. Please note, adding more than around 15-20 variables renders the chart unreadable.

 **Task:**

Look at the highest and lowest correlations with your dependent variable. Add all the media variables and view the Pearson’s correlation coefficient between some of the variables.

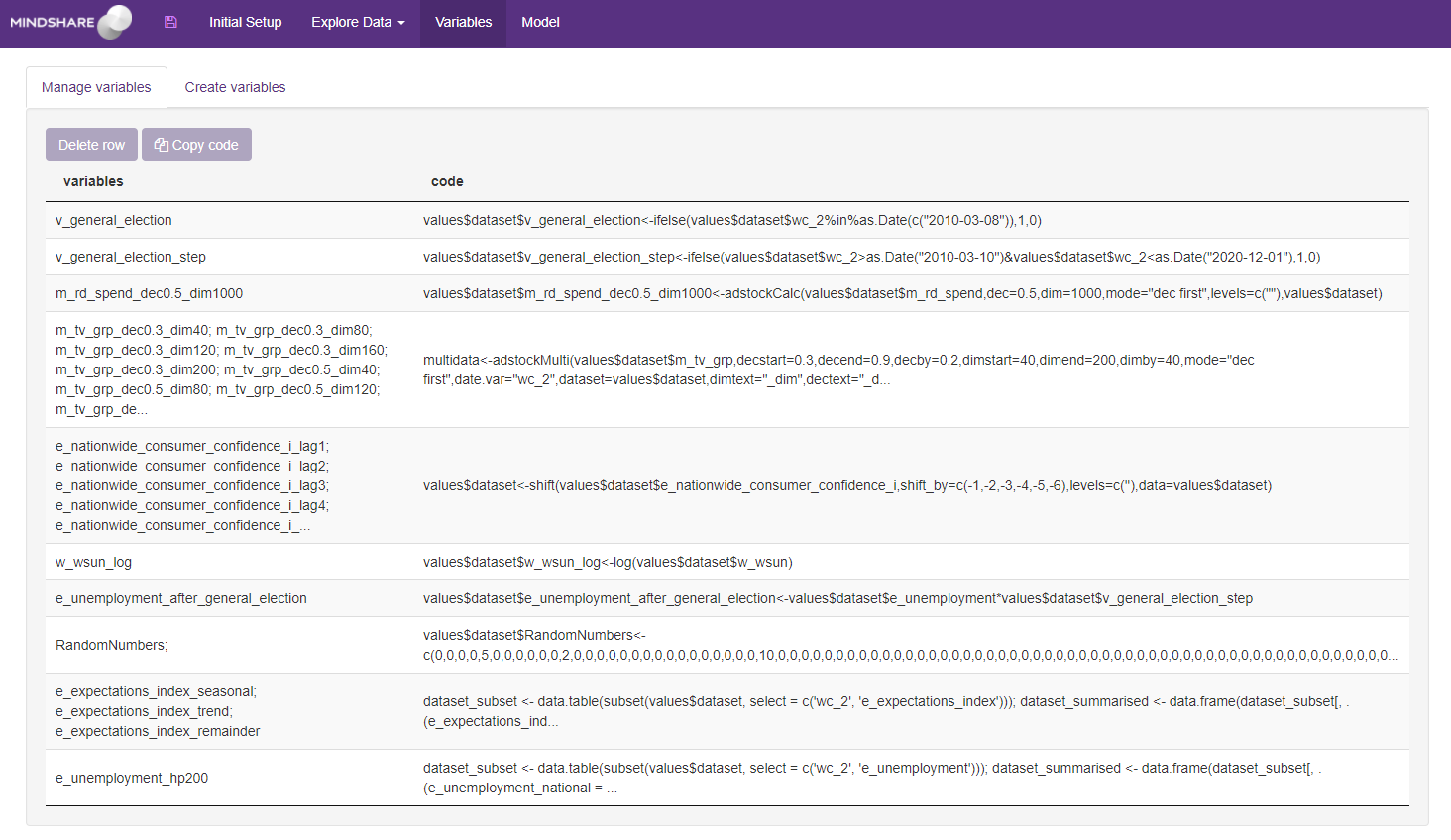
### Panel Data

The Panel Data tab should be used for panel level data. See [example 2](#_Panel_Data) for more information on this.

## Variables

The variables tab consists of a tab to manage created variables, and a tab to create variables.

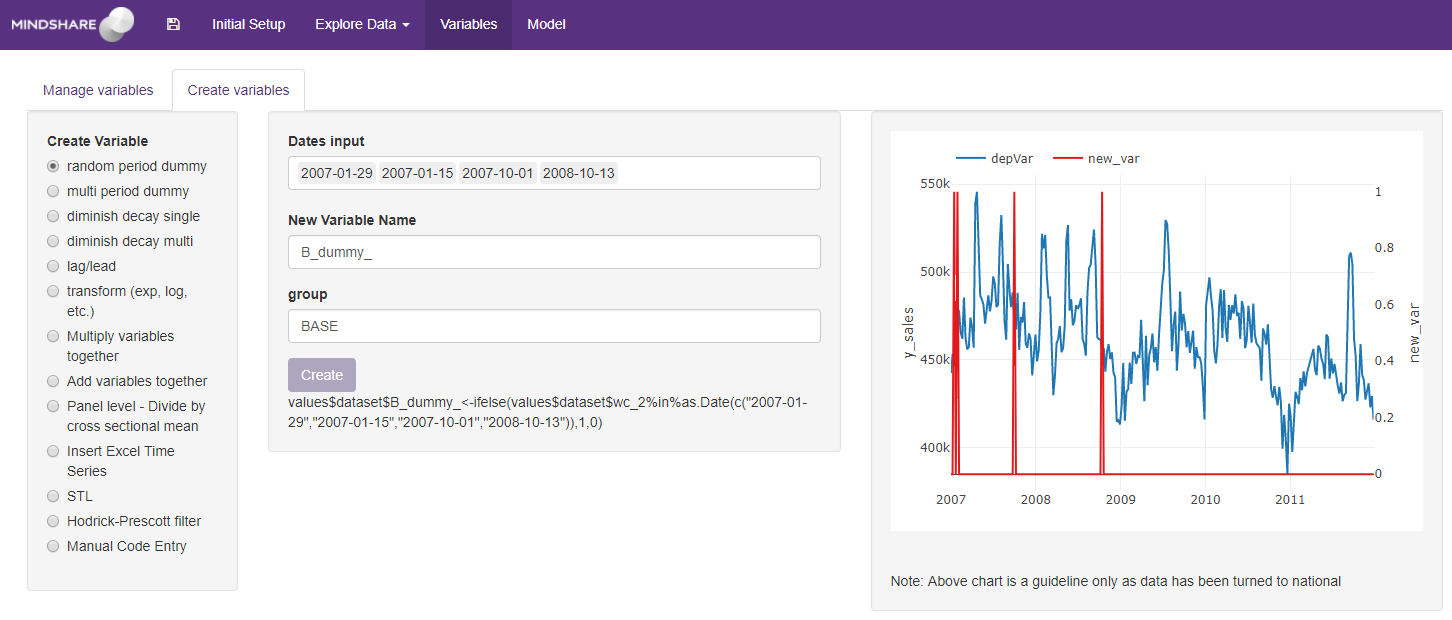
### Manage Variables



The manage variables tab is a table that contains any variables that get created in the Create Variables tab. From here, you can copy code from a previous line of created variable, or delete the code and its related variables from the dataset.

When reloading an old model (through a .lmmm file), the variables are already saved in the dataset as their time series, so the code won’t get re-run – However, the log of which variables have been created still exist in this table.

### Create Variables



The create variables tab is used to create any variables without having to revert back to writing code. The left hand drop down shows the possible variables that can be created. These are:

* Random Period Dummy
* Multi Period Dummy
* Diminish Decay Single
* Diminish Decay Multi
* Lag/Lead
* Transform (exp, log, etc.)
* Multiply Variables Together
* Add Variables Together
* Panel Level – Divide by cross sectional mean
* Insert Excel Time Series
* STL
* Hodrick-Prescott filter
* Manual Code Entry

The code for the current variable creation can be seen at the bottom centre of the page. Whenever the create button gets pushed, this code gets written out to a varCreate.R file that is saved wherever the save location has been chosen on the Initial Setup page. The code also gets run, creating the variable in the current buildModel.

If the shiny application is stopped the variable will not appear when rerunning buildModel (To combat this, you will have to run source(varCreate.R) to run any variable creation steps. It is worth managing the varCreate.R file regularly to avoid long run times, and variables that are no longer required.

### Random Period Dummy and Multi Period Dummy

Choose the dates you want to create a dummy for, create a variable name other than the default and choose the group you want the variable to sit under. Click Create.

Use the random period dummy to create a variable called V\_General\_Election. The dates for this are 3rd May 2010. Put this in the group called ‘Events’

Use the Multi Period Dummy to create a variable called S\_year2010. Select the start date as 01 Jan 2010 and the end date as 31 Dec 2010. Put this in a group called ‘Seasonality’

 **Task:**

Use the random period dummy to create a variable called v\_general\_election. The dates for this are 3rd May 2010. Put this in the group called ‘Events’

Use the Multi Period Dummy to create a variable called v\_general\_election\_step. Select the start date as 3rd May 2010 and the end date as 31 Dec 2020. Put this in a group called ‘Events’

### Diminish Decay Single and Multi (adstock)

There are two options for creating diminishing returns, creating a single variable, or creating a range. Either way has the option to create them by diminishing first, or by decaying first. At Mindshare we decay first[[2]](#footnote-2). Setting the diminish to zero or the decay to 1 will exclude that step.

These 2 functions have become defunct now you can test adstock in your model directly (saving space and hence time [see [Test Diminish](#_Test_Diminish)])

 **Task:**

Use the Diminish Decay Single function to transform the variable m\_rd\_spend with a decay of 0.5 and a diminish of 1000. Run this as decay first. There are no levels in this dataset so leave that blank.

Use the Diminish Decay Multi to transform the variable m\_tv\_grp decay first, with diminishes from 40 to 200 by 40, and decays from 0.3 to 0.9 by 0.2. There are no levels in this dataset so leave that blank.

### Lag/Lead

It is possible to create multiple lag and leads in one line of code. Use negative values for lags and positive values for leads. Separate multiple lags with a comma

 **Task:**

Lag the variable e\_nationwide\_consumer\_confidence\_i by 1 to 6 weeks (Use negative values for lags). There are no levels.

### Transform (exp, log etc.)

This creates a number of standard transformations to a variable.

 **Task:**

Transform the variable w\_wsun with the natural log transformation.

### Multiply/Add variables together

Here you can select variables you want to either add or multiply together, creating a new variable.

 **Task:**

We suspect there may be a change in how unemployment is measured after the general election in 2010. Multiply e\_unemployment by v\_general\_election\_step. Call this variable e\_unemployment\_after\_general\_election and add it to the group Economy.

### Panel Level – Divide by cross sectional mean

Not required for national level models. See [Example 2](#_Panel_Level_–) for an example.

### Insert Excel Time Series

Here you can insert data copied directly from excel. It is possible to change the names of the variables using the box at the top. Usual excel commands can be used such as

* Copy and paste
* Ctrl+enter to fill in multiple values at the same time when highlighted
* Drag to fill

 **Task:**

Enter some random numbers into var1, calling the variable RandomNumbers to get a sense for how the table works. Add this variable to your dataset.

### STL (Seasonal and Trend decomposition using Loess)

STL is a method for decomposing time series into the following components: seasonality, trend and the remainder.

 **Task:**

Apply STL decomposition to the variable e\_expectations\_index. Apply this at a national level.

### Hodrick-Prescott filter

The Hodrick-Prescott filter is a tool used to remove the cyclical component (i.e. extract a trend component) of a time series. It is mostly used to obtain a smoothed-curved representation of a time series. The smoothness is achieved by specifying a multiplier λ (default is 1600).

 **Task:**

Apply the Hodrick-Prescott filter the variable e\_unemployment. Choose either 200, 2000 or 20000

### Manual Code Entry

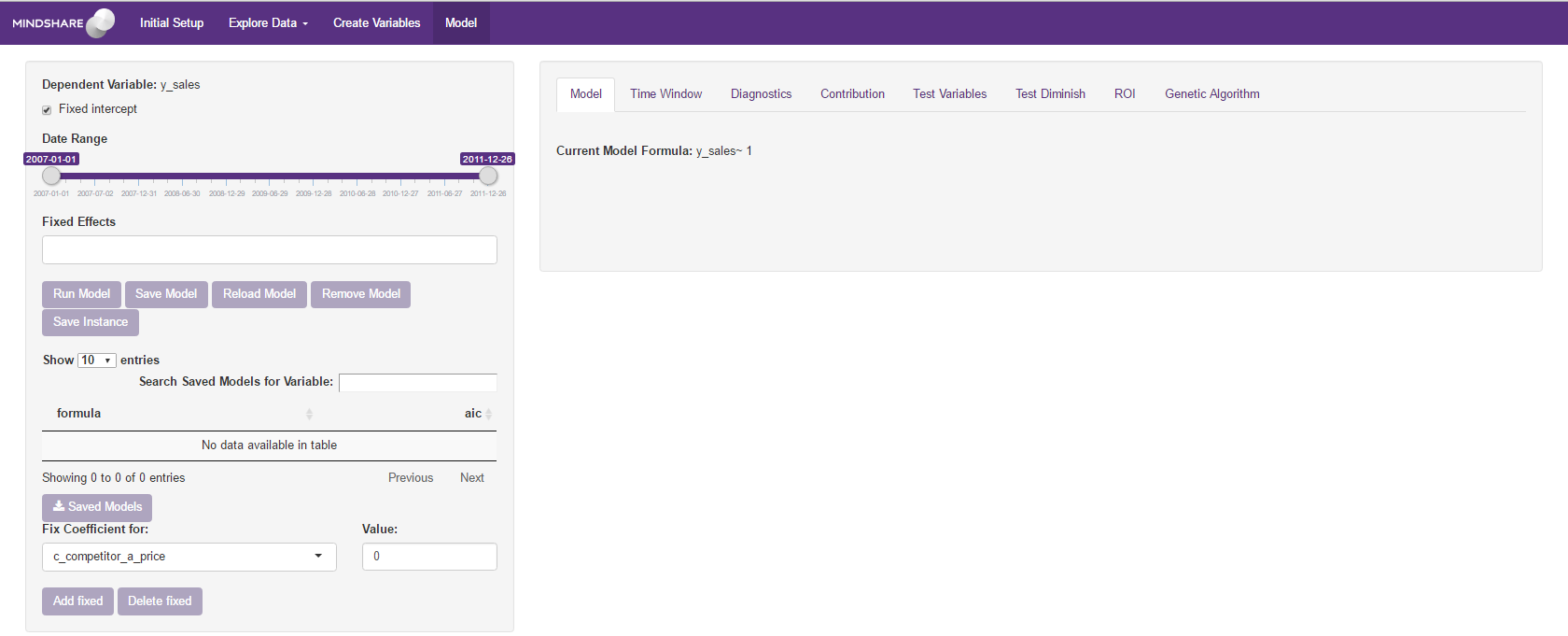
The Manual Code entry allows the user to enter any code they wish to create variables. There are two boxes to fill information in

* **Variables that get created**: Name every variable that gets created, split by semi-colons (;). If you don't specify all the variables that get created, you won't be able to delete those variables from the manage variables tab
* **Code**: Enter the code to create the variables in here. To add a variable, add it to values$dataset. E.g. to create a column of ones called 'test', you would run the following line of code: values$dataset$test<-1

 **Task:**

Copy the code for one of the previous variable creations, add it to the second box and adjust the parameters to your choosing. Add the variables it creates into the top box.

## Model

The model tab is the heart of the buildModel function. It is split into two sections

* An always on left hand side showing model variables and previous models
* A tabset right hand side showing various information relating to the current model. This is made up of 8 separate tabs which will be explained below.

### The left hand side

1. The left hand side remains throughout the entire model tab. Variables in the buildModel formula will automatically populate the tick boxes and variable boxes on this page. The number of tick boxes and variable boxes depends on the number of mixed levels selected on the original page (see [example 2](#_Example_2:_Using)). The fixed intercept and fixed effects variable box will always exist.

A slider exists to change the date range that the model will run for.

The run model button will run a model with the specification explained by the variables in the boxes above.

1. It is possible to save one of these models for later use. The 3 pieces of information that get saved are the model formula, the AIC, and the optimiser used (if mixed model).

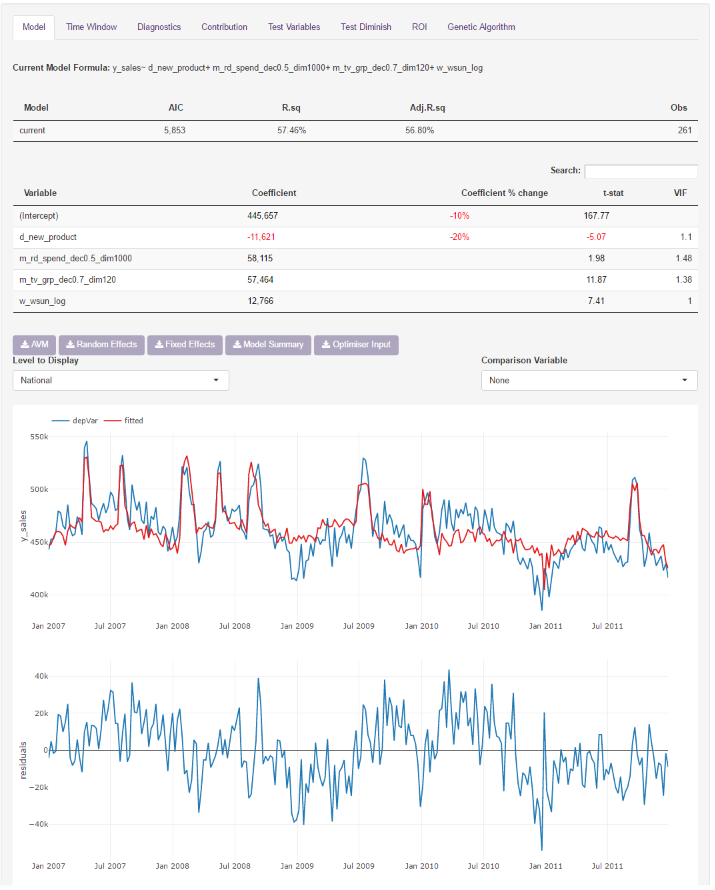
Once models have been saved, the first table will appear on the LHS with the model formula and it’s aic. This contains any saved models that may want to be referred to, used, or compared at a later stage. Clicking on a model in this table will run that model as a comparison, and provide statistics for both models.

There are buttons to reload the model (Move all the variables in the list into the variable boxes above), remove a model (delete’s the row from the savedModels table).

1. There is an option to fix coefficients in the model. Choose the variable and value you would like a coefficient to take and click “Add fixed”. By clicking on a variable in the table below, you can either edit the coefficient it takes, or delete that variable.

### The right hand side

#### Model



The model tab shows standard modelling outputs such as fit statistics, variable coefficients and variance inflation statistics (for multicollinearity). Coefficients also get tracked from one model to the next. There is a chart for the AVM as well as one for residuals vs a comparison variable. These can be drilled into for specific levels to view fit within.

There are several options for downloading outputs. AVM, random effects, and fixed effects download as expected. The model summary downloads a file containing everything from the dataset and coefficients, to contributions and ROIs, useful as a model closing file. The optimiser input download contains everything for running optimisations in the format required.

 **Task:**

Select the following variables, some of which will have been created by you, in the fixed effects:

* m\_rd\_spend\_dec0.5\_dim1000
* m\_tv\_grp\_dec0.7\_dim120
* d\_new\_product
* w\_wsun\_log

Click Run Model. You should see fit statistics, a coefficient table and actual vs model appear. Click the “Save Model” button the LHS (A table on the LHS should appear).

Add the variable p\_price to the model and run again.

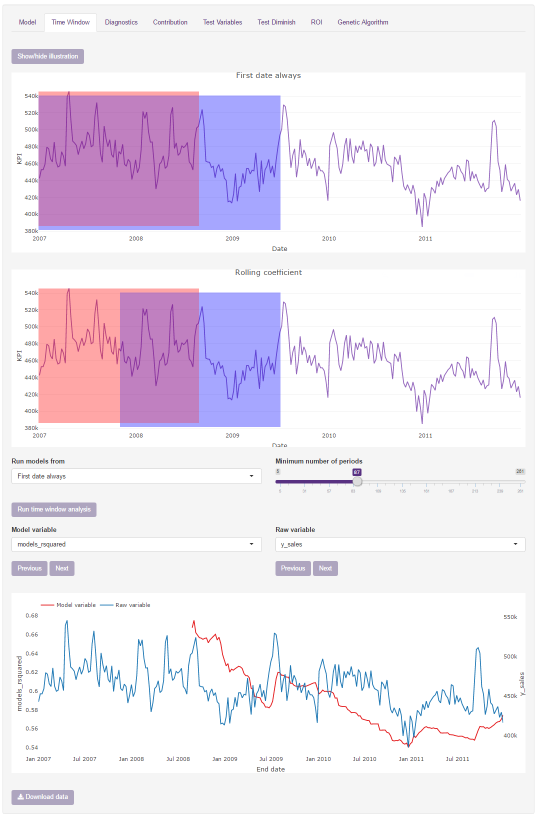
Click on the first saved model on the right hand side. Notice how the coefficient table on the Model tab changes, and you can compare the two models.

Unclick the comparison model.

Download the Model Summary.

Scroll through the rest of the tabs as the document progresses.

#### Time Window



The Time Window tab allows the user to get a view on how their model coefficients vary over time.

**Show/hide illustration button** – This button shows us how the time window feature works.

**Run models from dropdown**

1. First date always. This will fit a number of models (depending on the number of periods) always starting from the first observation. For example, if the period is 3 and the total number of observations is 10, then the first model will run from 1 to 3, the second model will run from 1 to 4, third from 1 to 5 and so on until we run out of data points.
2. Rolling coefficient. This is similar to the 'First date always' option except for one thing. Say that the period is again 3 and we again have 10 observations in our dataset. The first model will run from 1 to 3, the second model will run from 2 to 4, the third from 3 to 5 and so on until we run out of data points.

If 'Rolling coefficient' is selected, a dropdown called 'Display chart x-axis as' appears with two options available:

* **Start date/date** – This option will chart the analysis results from the start date.
* **End date/date** – This option will chart the analysis results from the end date.

**Minimum number of periods slider** – From here you can select the length of your period. The bigger your period the less models it's going to fit.

**Run time window analysis button** – Once the analyst is happy with the above input, clicking this button will run the time window analysis. The result consists of an illustration of how the model coefficients change over time. The of each model is also saved and can be selected from the 'Model variable' dropdown.

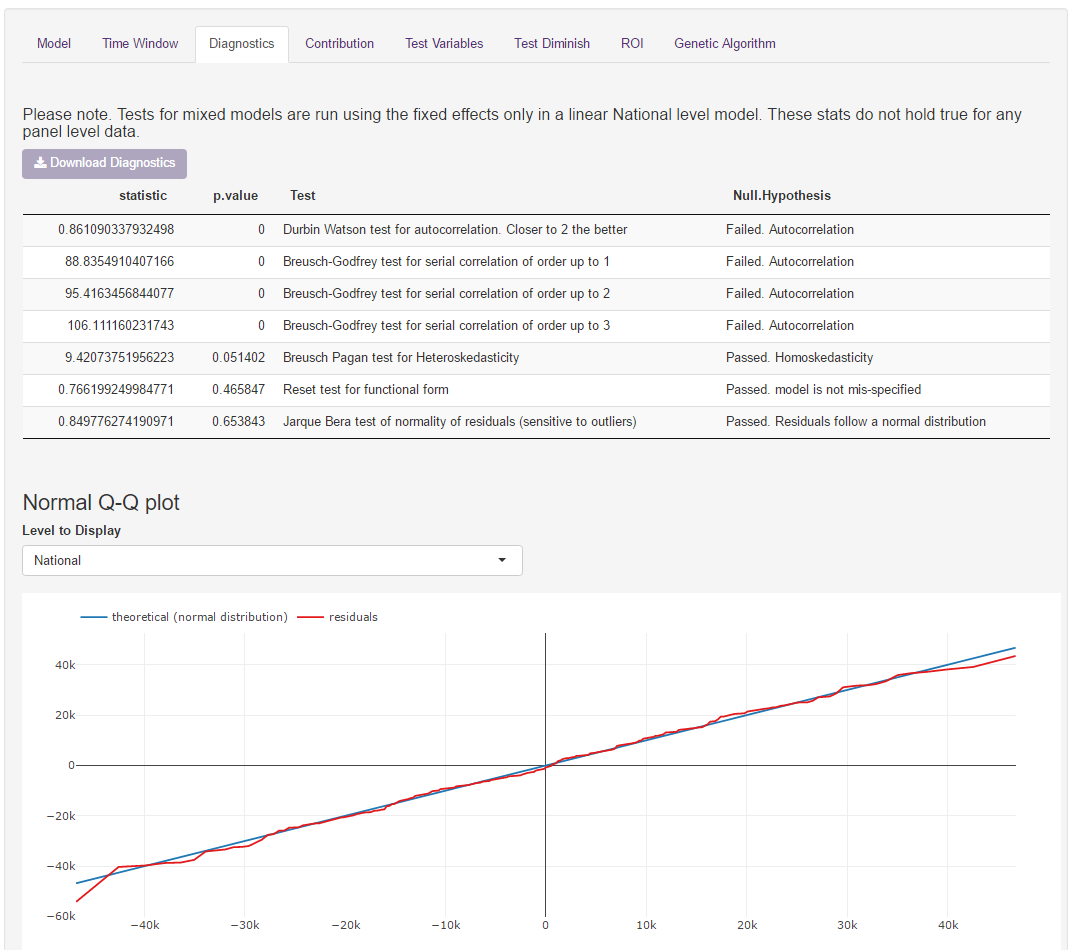
**Download data button** – This does exactly what you would expect, that is, it saves all the data to a CSV file.

 **Task:**

Select ‘Rolling Coefficient’ from the ‘Run Models from’ dropdown and then click ‘Run time window analysis’ again.

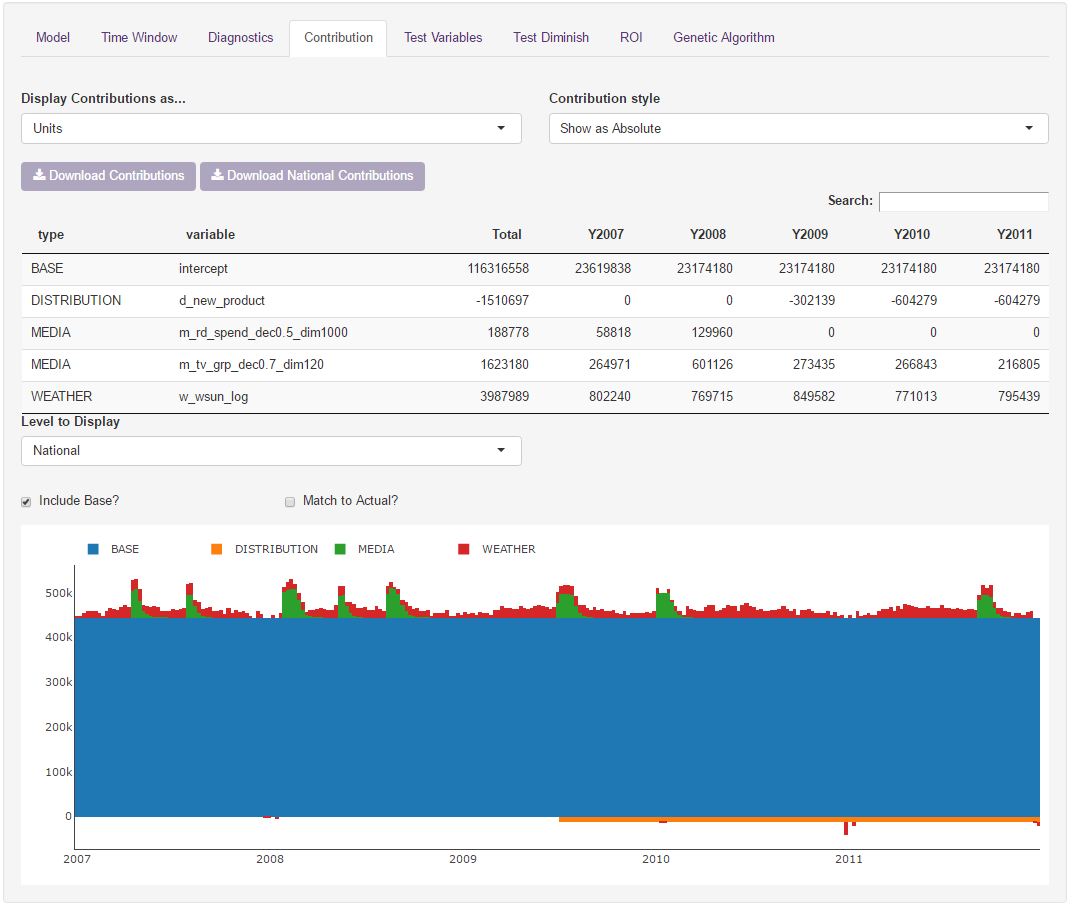
Change the number of periods and observe how the data is displayed.

#### Diagnostics



The diagnostics page shows linear model tests for the current model. Tests are shown as whether they have passed or failed the test. Moreover, a quantile-quantile plot is displayed.

#### Contribution



There are two main outputs on the contributions tab – a yearly summary and a weekly chart. The yearly summary can be compared to a previous model, like on the model tab.

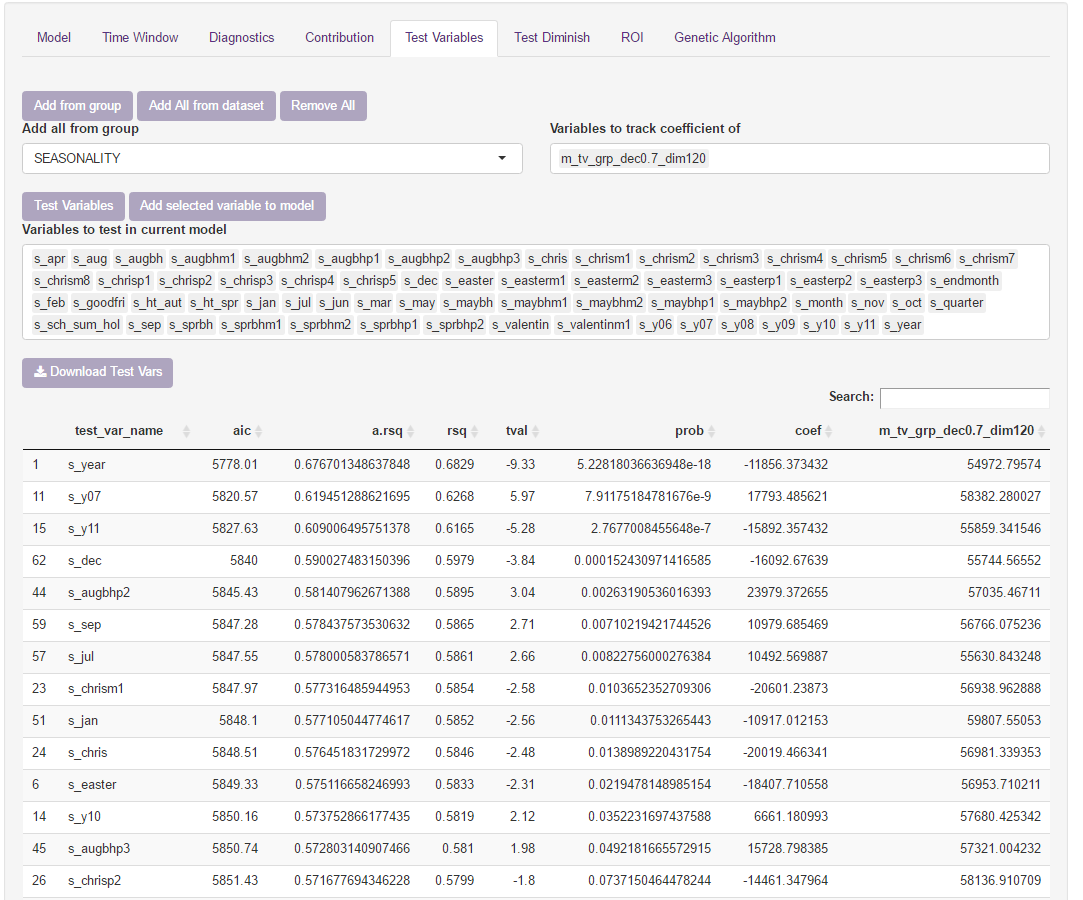
It is possible to view the contributions in their absolute value, by percentage of each period, or by percentage within each group (defined on the Initial Setup > Variable Groups page)

The contributions tab can be viewed in units or in value. When in value, the financial variable (chosen on the initial setup tab) is used.

It is possible to view the chart by variable or by group. Clicking on one of the variables/groups in the legend hides that variable from view.

Contributions can be downloaded as either national contributions or at their individual level. The output is a csv.

#### Test Variables



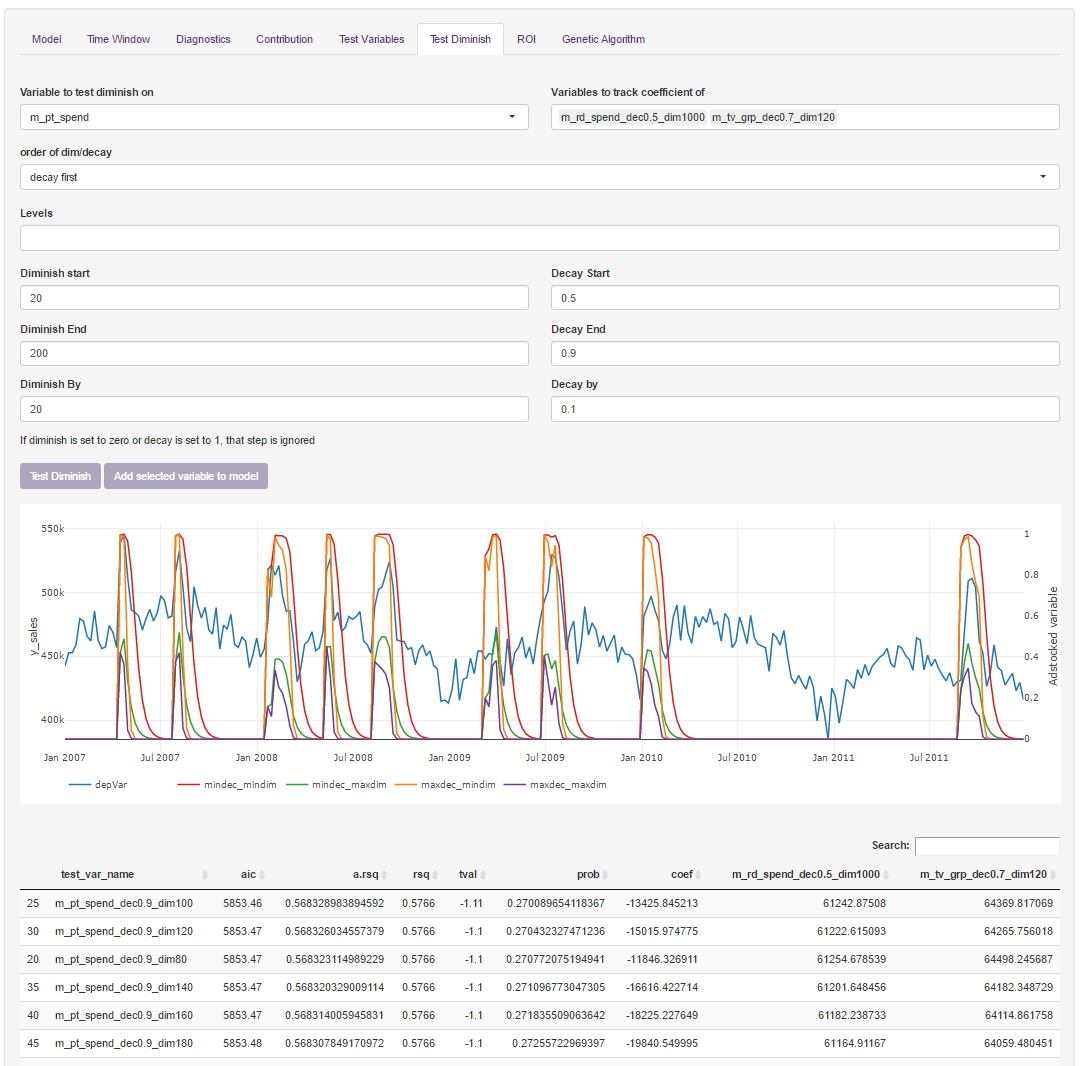
The test variables tab allows variables that aren’t in the model to be tested as a fixed effect one at a time. Once selected, the model loops through each of the variables, running the base model with one additional variable each time. This can be sorted by the metric of choice.

 **Task:**

Select Seasonality from the ‘Add all from group’ drop down and click add from group. Add the TV variable into the variable to keep track of.

Run test variables and sort by AIC. Add the top variable to the model if it makes sense.

#### Test Diminish



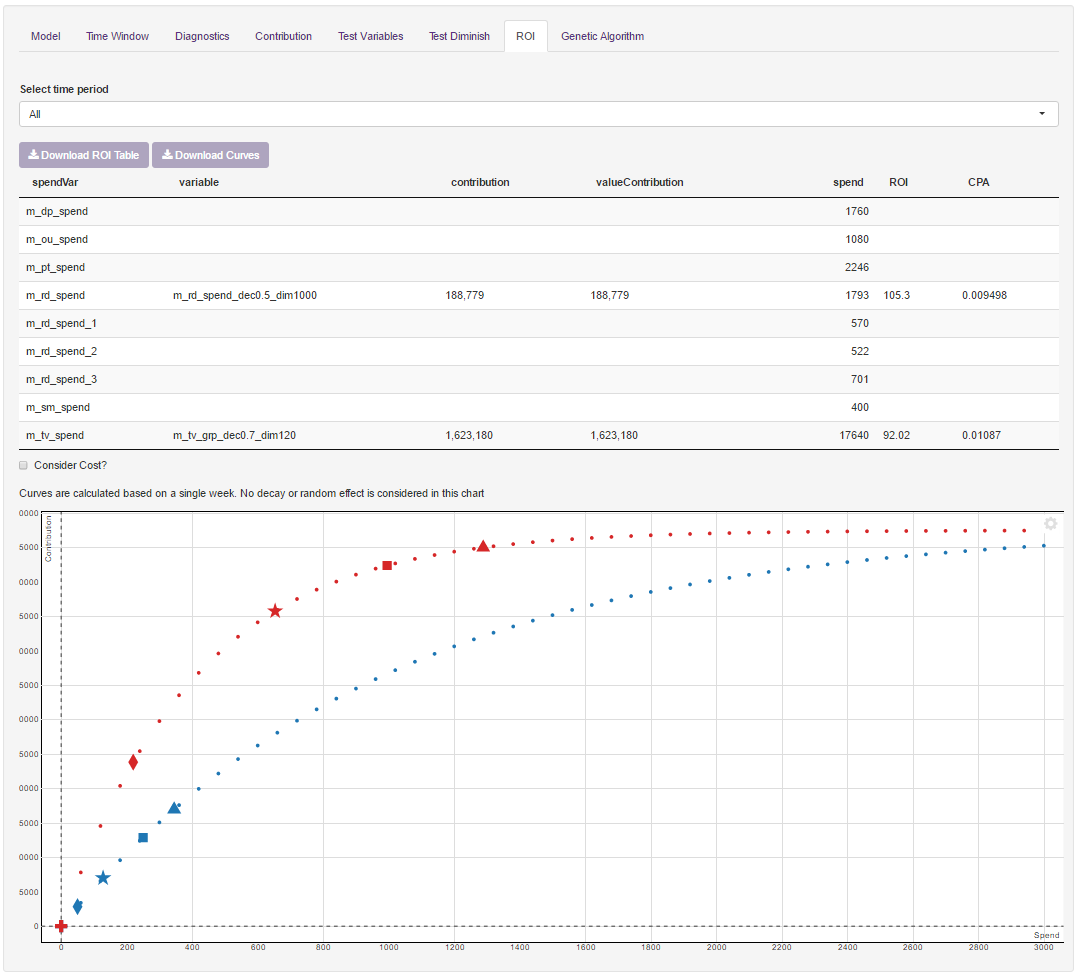
The test diminish tab has been built to replace the adstock creation in the create variables tab. Here you can test diminishes for one variable at a time, whilst keeping track of other variables that are currently in the model.

 **Task:**

Choose the m\_pt\_spend variable to test adstocks for. Test diminishes from 20 to 200 by 20, and decays from 0.5 to 0.9 by 0.1

Add a variable to the model if it makes sense.

#### ROI



Subject to variables being matched up to their spends in the Initial Setup tab, the ROI page will calculate a return on investment, and estimate a revenue curve (Note: Estimation of curves should be a guide only as this code makes some assumptions when grouping data up to the national level).

All spends are shown in this tab which allows you to view channels that you haven’t added to the model yet.

The points on the curve show the minimum (non-zero), average (non-zero), and maximum weekly spend.

##### Uploading to the ROI Database

Clicking on ‘upload to ROI database’ will bring up a pop-up for the user to fill in.

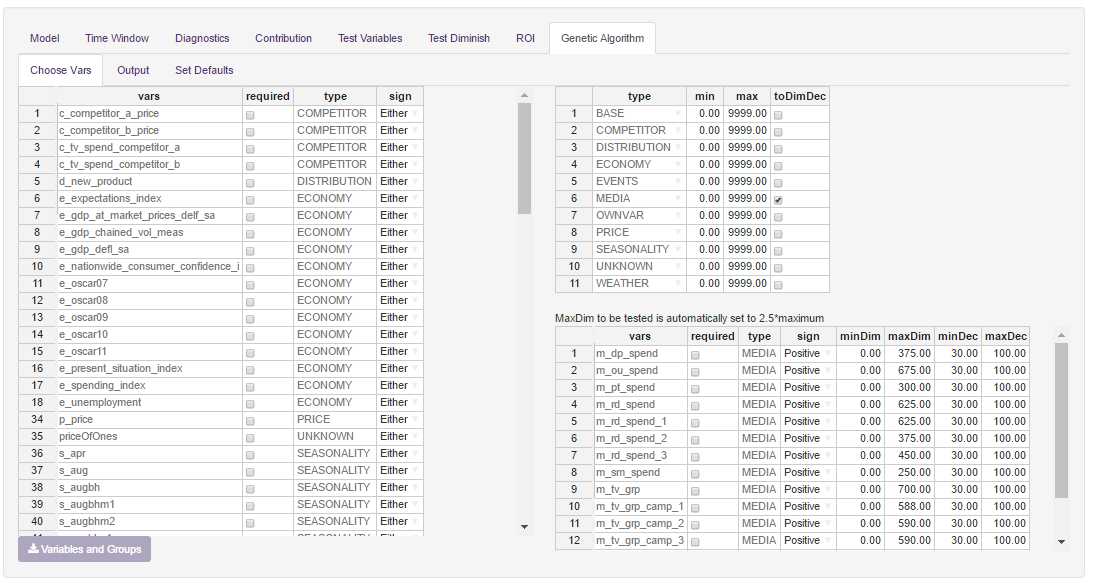
Campaign Name and Campaign Type are left blank for the user to fill in, but are required for any variables where Include is ticked.

## Genetic Algorithm

A genetic algorithm (GA) is an evolutionary algorithm, inspired by the process of natural selection. It automates model selection, deriving the model of best fit given certain parameters.

The 3 sub-tabs on the GA tab relate to setting up (Choose Vars), viewing the model output (Output) and adjusting the methodology of the GA (Set Defaults)

### Choose Vars

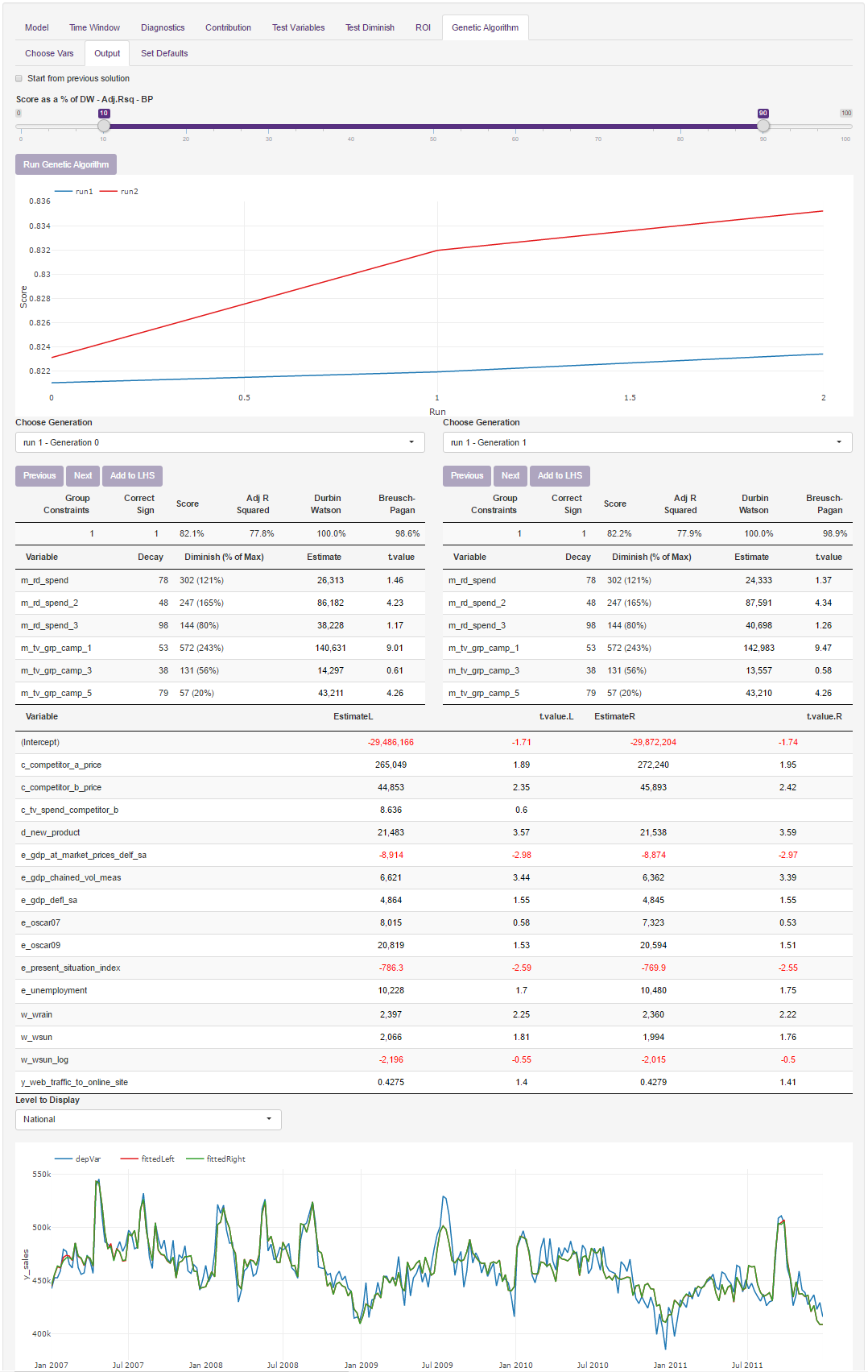


There are 3 tables on this page which help to set up the genetic algorithm

* **Group Table** – This table (top right) allows you to set the minimum and maximum number of variables that are in each group. It also allows you to decide which groups you would like to apply diminish and decays to.
* **Variable Table** – This table (left) shows variables that are not getting dim/decayed. You can decide whether you want a variable to be forced into the model or not, and change the sign.
* **Transform Table** – This table (bottom right) shows variables that have been chosen to be dim/decayed. MaxDim automatically sets to 2.5 times the maximum value of that variable.

Once you have chosen which variables to include, their signs and diminishes, move onto the Output tab.

### Output



The model gets run from this page. First, you must choose the breakdown of the score as a percentage of Durbin Watson (Autocorrelation), Rsquared (Model fit), and Breusch Pagan (Heteroskedasticity). This is the score that helps decide the best model. Once chosen, run your GA model.

The GA model takes time to run. On the test dataset (celeb.mod.trans) it takes between 5 and 14 minutes. You can check that the model is running in the R console.

Once run, you can compare two generations of models side by side. There are three tables and a chart:

* **Fit statistics** – This contains whether the model passed the group constraints (Min and max of each group), whether the model passed the sign constraints, and then the fit statistics.
* **Media Variables** – This shows the media variables as well as their chosen dim and decay, along with the coefficient.
* **Non Media Variables** – This shows the coefficients for any variables that are not diminished and decayed.
* **Actual vs Model chart** – This compares the fits from each model.

Once a model has been chosen, clicking “Add to LHS” creates any variables that need creating and adds them to the fixed variables. One is then able to

### Set Defaults

The set defaults tab is for those that want to fine tune their genetic algorithm. Here it is possible to adjust the parameters that build the model. For a full list of these parameters see ?rgenoud for more details.

 **Task:**

Go to the Choose Vars page. Make e\_expectations\_index a required variable. Change c\_tv\_spend\_competitor\_a and c\_tv\_spend\_competitor\_b to have a negative sign. Make m\_tv\_grp required also.

Go to the set defaults tab and set Max Generations to 30, and tick the Hard Generation Limit (This means that only 30 iterations will be run, speeding up the demo)

Go to the output page. Set the DR-Adj.Rsq-BP relationship to be 20%-70%-10% and click run. Compare a generation from Run 1 with a generation from Run 2. Pick one of these models and add to LHS. Click Run Model on the LHS and return to the model tab to see your new coefficients.

# Example 2: Using lmmm’s built in UI (single equation) for a panel level or mixed effects model

As well as being able to run linear models, the buildModel function can run panel level or mixed effects models. There are few additional details on top of example 1 that are detailed below.

## Initial Setup

These additions should be present in a panel or mixed effects model

**Mixed Levels:** If your data is not at a national level, then this is required. It should detail the variable names that explain each of the levels. These can be overlapping or independent. For example it is ok to have the following: *store\_number, region, store\_type, and product* where region and store\_type are nested within store\_number (ie there may be multiple stores in a region), product is separate to store\_number. The major rule is that there should not be multiples of the same date for each exclusive level, ie. store\_number 1, product = widgets. Also, having a subset of dates for a level will cause the dependent variable to be mean divided

To automatically assign these in the model formula they should be in brackets, separated by a bar (|) e.g. y ~ x1 + x2 + (x3|store\_number) + (x4|product). It is recommended to have as many options in the mixed levels as possible where you may in the future want to test something at this level.

**Correlation of Mixed Effects**

The correlation of mixed effects is an option to decide if you want to create a covariance matrix for each of the variables in your mixed effects. The covariance matrix details the similarity in the random effect coefficients. A high value means a likely change in coefficients given one other coefficient changing.

Uncorrelating the mixed effects reduces the complexity of the model, so it may fit quicker. However, Models in which the slopes and intercepts are allowed to have a non-zero correlation are invariant to additive shifts of the continuous predictor. This invariance breaks down when the correlation is constrained to zero; any shift in the predictor will necessarily lead to a change in the estimated correlation, and in the likelihood and predictions of the model.[[3]](#footnote-3)

**Restricted Maximum Likelihood**

when you defined REML = FALSE you used the Maximum Likelihood estimated instead of the Restricted Maximum Likelihood one. The REML estimates try to "factor out" the influence of the fixed effects XX before moving into finding the optimal random-effect variance structure (see the thread "[What is "restricted maximum likelihood" and when should it be used?](https://stats.stackexchange.com/questions/48671)" for more detailed information on the matter). Computationally this procedure is essentially done by multiplying both parts of the original LME model equation y=Xβ+Zγ+ϵy=Xβ+Zγ+ϵ by a matrix KK such that KX=0KX=0, i.e. you change both the original yy to KyKy as well as the ZZ to KZKZ.[[4]](#footnote-4)

**Mixed Effects Optimiser**

This specifies the optimiser that the lmer function (from lme4 package) will use to estimate the model. The gold standard would be to try all available optimizers, which, whilst being slow for large fits will provide a view as to whether the model is stable or not. If all optimizers converge to values that are practically equivalent (it’s up to the user to decide what “practically equivalent means for their case”), then we would consider the model fit to be good enough.[[5]](#footnote-5)

 **Task:**

Clear the workspace, load up the dataset celeb.mod.panel and run buildModel on this dataset.

Rm(list=ls())

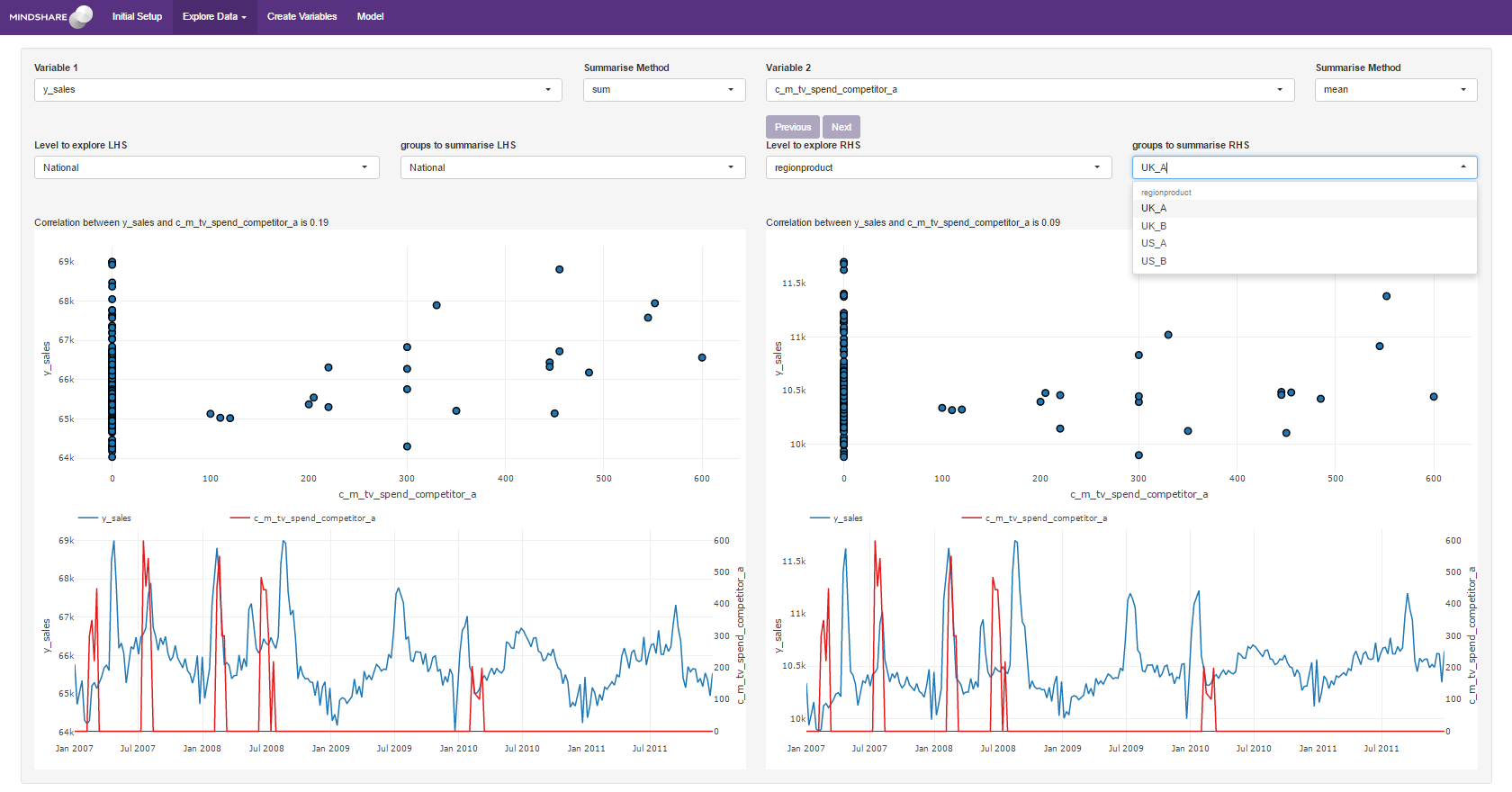
data(celeb.mod.panel)

buildModel(celeb.mod.panel)

* Set the dependent variable to y\_sales
* Set the mixedLevels to region, product, and regionproduct
* Set the dependent transformation to “mean divided”
* Set the price variable to p\_price
* Set the date variable to be wc\_2

## Explore

### Panel Data



The Explore tab is best used when using panel level data. It is possible to view 2 variables for 2 regions side by side. In the example above we are comparing our dependent variable y\_sales (in blue) against competitor spend (in red) for National sales (on the LHS) vs Product B (on the RHS). The three outputs for each side are a Pearson’s correlation, a scatter plot and a line graph. All plots are built using Google Charts, so internet connectivity is required to view the charts.

 **Task:**

Compare y\_sales vs c\_m\_tv\_spend\_competitor\_a for the two regionproduct’s UK\_A and UK\_B. Notice the different correlations between the two variables for the two regions.

## Var Create

If running a model that is not at national level, the levels needs to be an exhaustive list of the levels that you want to diminish and decay by – There should not be multiples of the same date for these levels.

 **Task:**

Create a diminish decay single for m\_rd\_spend with a diminish of 1000 and decay of 0.5. Make sure that the level regionproduct is in levels (region and product can be in there, but are superfluous).

### Panel Level – Divide by cross sectional mean

This function is mostly for panel level data where you want to apply a transformation to a variable, dividing by the variable’s mean. This brings the variable in each region around an average of 1. The added benefit is that the coefficient works closer to a percentage: A 1% increase in this variable leads to [coefficient] increase in the dependent variable.

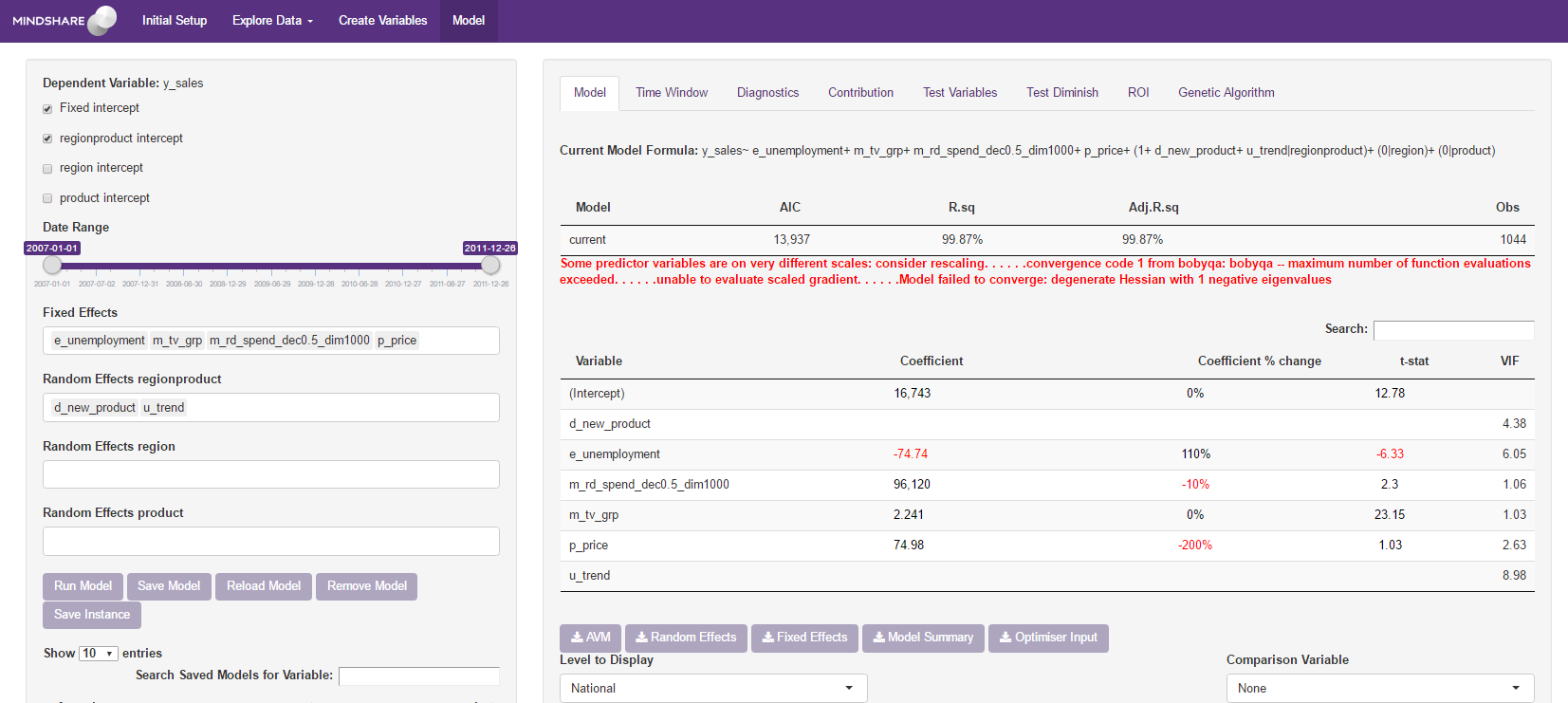
A strong use case for mean dividing is when there are large variations in the variable between regions, and a unit increase in the variable leading to a [coefficient] increase in the dependent variable doesn’t make sense.

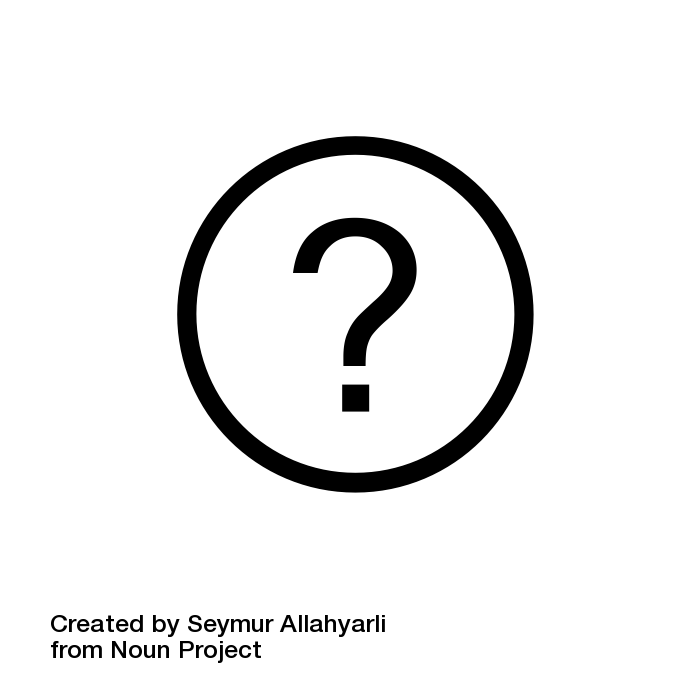
 **Task:**

Mean divide the variable e\_unemployment.

## Model

When mixed levels are added in the setup page, there are additional boxes on the left hand side of the model page relating to mixed effects for each of the intercepts and variables. When adding variables into the mixed effects



 **Tip:**

The lmer package that gets used for mixed effects modelling generally comes up with a number of warnings, and a fear that the model failed to converge. The gold standard would be to try all available optimizers, which, whilst being slow for large fits will provide a view as to whether the model is stable or not.

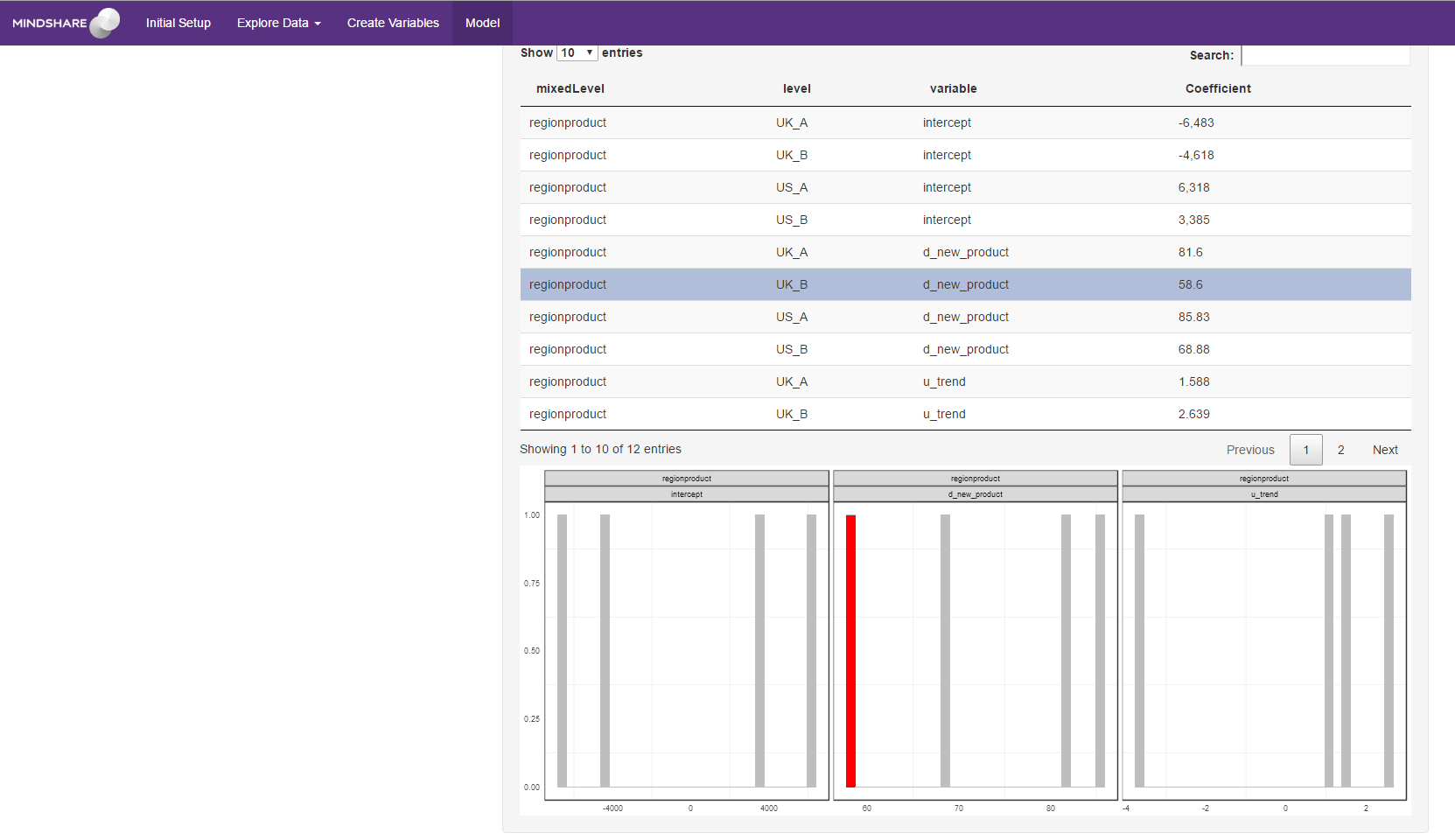
 **Task:**

Allow the intercept to vary by regionproduct

Add the variables e\_unemployment, w\_wtemp, m\_tv\_grp, m\_rd\_spend\_dec0.5\_dim1000, p\_price as fixed effects

Add the variables d\_new\_product, and u\_trend to the random effects by regionproduct

Run the model.



Two additional boxes appear towards the bottom. These are related to the mixed effects component of the model. The first is a table detailing each of the coefficients. This can be downloaded by click on the Random Effects download box. The second is a histogram showing the spread of coefficients for each random effect. Clicking on a row in the table will highlight that data point on the histogram.

When running the time-window analysis, only the fixed coefficients are saved and charted by the model. Adding random coefficients may be a feature in future updates.

When running a genetic algorithm, only fixed coefficients get tested. Adding random coefficients may be a feature in future updates.

# Appendix A – Changing reference points

Reference points change little when running a linear model, however they have a significant impact when running a log model. Using an example of price (non-zero for the entire time) where price starts at £1 and rises to £1.10 half way through the modelling period. Assume a negative coefficient of -2.

* Keeping price as it is: When looking at a contribution chart, price will likely be a large proportion of sales, whilst the intercept will be significantly lower than you would expect. This is because a linear model looks at the variation in a variable rather than the actual variable itself, and the model is creating the price coefficient based on the change from £1 to £1.10, not on the fact that the price is £1 or £1.10.
* Removing any number from the price variable will make the contribution of the intercept larger and more in line with what is expected.
* Removing the minimum from the price variable (before modelling) will mean that the variable is 0 until half way through, then 0.1 for the second half. A negative coefficient will show a negative contribution in the second half from the higher price.
* Removing the maximum from the price variable (before modelling) will mean that the variable is -0.1 until half way through, then 0 for the second half. A negative coefficient will show a positive contribution in the first half from the lower price.

Ultimately, all of the above changes will show the same contribution change when going from £1 to £1.10.

When modelling log sales, the reference point of the contributions will matter a lot more. Having a strong negative price will likely lead to having a higher intercept value. All calculations carried forward will be calculated off this higher base value, leading to larger contributions (Assuming the intercept could have an coefficient of either 2.6 or 1.6, and a variable has a coefficient of 0.1, exp(2.7) – exp(2.6) is much bigger than the exp(1.7) – exp(1.6)).

Also to take note is that for price, the share of the leftover contribution to split will be much bigger if no min or maxing is run.

# Appendix B - Adstocking

Regarding the diminish parameter, one should test values from around 5% of maximum to 300% of maximum in reasonable steps, and in the model see which provides the best model, and then test around a smaller range.

For example, if GRPs goes up to a maximum of 100, A diminish of 10 means anything above 40 GRPs is completely wasted (no extra gain) whilst a diminish of 200 looks pretty linear up to 100 GRPs, and then starts to see diminishing returns.

The decay will also need to be considered in this calculation. By decaying first and have a decay rate of 10% (90% carryover), with four weeks of 100 GRPs the value you will be diminishing is 100 + (100)\*0.9 +100\*0.9^2 +100\*0.9^3 = 344 GRPs on the fourth week, rather than 100. In this case it may make sense to test diminishes beyond 300% of the maximum 100 GRPs, perhaps up to 900.

When creating the decayed variable in the Create Variables tab, there is a graph on the right hand side that should help as well, which shows the extremes of the variables that you are creating (“minimum diminish & minimum decay” through to “maximum diminish & maximum decay”).

# Appendix C – Log Decomposition explanation

There are a number of different ways to calculate log decompositions. Below is the agreed Mindshare methodology:

1. Calculate the contribution for the BASE group
2. Based on the groups on the setup page, calculate the contribution for each group based on exp(BASE + group) – exp(BASE). Repeat for each group
3. Split the difference between exp(BASE + all groups) and sum(exp(BASE +group)) between each group according to the weekly weighting
4. Within each group, calculate the contribution for each variable as exp(BASE + variable) – exp(BASE). Repeat for each variable in one group. Split the remaining contribution for that group between the variables according to the weekly weighting.
5. Within the BASE, contributions for each variable are structured as exp(intercept + variable) – exp(intercept)

# Appendix D – Dealing with Collinearity

Econometric literature advises on two “practical” ways of dealing with collinearity:

1. Omit the collinear variable from the model, if it’s feasible.
2. Transform the collinear variable, to reduce the collinearity.

The problem arises when you cannot omit the collinear variable for example:

Your client is an FMCG in a growing market, has revenue, volume sold and price upward trending. Your price variable is highly collinear with the trend variable you have capturing market growth, but you need both in there. If you omit the market growth trend, you end up with a positive coefficient on price, due to the upward trend – suggesting model misspecification. What would you do in this case?

As the objective of a multivariate regression is to calculate the coefficient of each driver, with all other drivers being controlled for. What you want to try is to find the true beta for each of the collinear variables. You can do this by running the regression with each variable at a time and observing the coefficients. Then run the regression once again with both collinear variables and observe the coefficients – Did anything jump? By how much? Can you live with it? What’s their level of significance? Do the variables pass the F-test of joint significance in the regression?

If the coefficients don’t jump, that implies that despite the collinearity the variables are stable and you are ok with having both in. If they do, then it’s back to the drawing board trying to find a transformation for price or distribution which will resolve the collinearity. Some useful transformations you can test are:

* Logs.
* Relative price.
* Segregate a “base” and “promotional” price.

Media variables will almost always have some level of collinearity, due to the nature of our business a lot of media will be planned in conjunction to each other. Resolves for collinearity with media include:

1. Modelling spends and combining media variables.
2. Testing different levels of decay.

In the occasion where your media variables are not heavily collinear, but do make each other less significant. Observe the coefficient of each variable, and their coefficients when they are both in the model and make an assessment whether the collinearity is causing bias or instability. Then run an F-test of joint significance of the variables to satisfy statistical robustness.

# Appendix E - Notes on Mixed Effects

There are two ways to estimate a panel level model. The first is Dummy Variable Ordinary Least Squares. The second is through a mixed effects model. In the DVOLS model you are fitting fixed effects parameters to model (one for each group) whilst in the Mixed effects model you are saying the random effects have a normal distribution of which you are estimating the variance (and also therefore the adjustments to coefficients). For reference, MarketingQED generally uses a Dummy Variable OLS model, whilst eViews uses mixed effects models.

If running a mixed effects model, there are several additional factors that must be taken into account, detailed below.

## fixed and random effects with REML

If running a model without first demeaning your variables, you are required to add the variable to the fixed effects as well as the random effect. This is because the mixed effects function works similar to a 2 stage approach – first estimating the fixed effects, and then estimating the variability around that fixed effects with the random effects. This is not the case with running a Dummy Variable OLS model (DVOLS) but will not cause it to break (REML true/false doesn’t matter for DVOLS).

When defining REML = FALSE you used the Maximum Likelihood estimated instead of the Restricted Maximum Likelihood one. The REML estimates try to "factor out" the influence of the fixed effects XX before moving into finding the optimal random-effect variance structure (see the thread [What is "restricted maximum likelihood" and when should it be used?](https://stats.stackexchange.com/questions/48671/what-is-restricted-maximum-likelihood-and-when-should-it-be-used) for more detailed information on the matter). Computationally this procedure is essentially done by multiplying both parts of the original LME model equation y=Xβ+Zγ+ϵy=Xβ+Zγ+ϵ by a matrix KK such that KX=0KX=0, i.e. you change both the original yy to KyKy as well as the ZZ to KZKZ.

To estimate a variable at the random effect, it is suggested you should have at least 6 IDs at the random level, however, if you have less then 6 your model simplifies to a classical linear model, which is fine.[[6]](#footnote-6)

## The Optimiser, and “failures to converge”

The lmer package (lme4) that gets used for mixed effects modelling generally comes up with a number of warnings, and a fear that the model failed to converge. The gold standard would be to try all available optimizers, which, whilst being slow for large fits will provide a view as to whether the model is stable or not.

If the model is not stable, you should try to simplify your model.

## Should we even be running a mixed effects model?

Several considerations will affect the choice between a fixed effects and a random effects model.[[7]](#footnote-7)

1. What is the nature of the variables that have been omitted from the model?
   1. If you think there are no omitted variables – or if you believe that the omitted variables are uncorrelated with the explanatory variables that are in the model – then a random effects model is probably best. It will produce unbiased estimates of the coefficients, use all the data available, and produce the smallest standard errors. More likely, however, is that omitted variables will produce at least some bias in the estimates.
   2. If there are omitted variables, and these variables are correlated with the variables in the model, then fixed effects models may provide a means for controlling for omitted variable bias. In a fixed-effects model, subjects serve as their own controls. The idea/hope is that whatever effects the omitted variables have on the subject at one time, they will also have the same effect at a later time; hence their effects will be constant, or “fixed.” HOWEVER, in order for this to be true, the omitted variables must have time-invariant values with time-invariant effects.
      1. By time-invariant values, we mean that the value of the variable does not change across time. Gender and race are obvious examples, but this can also include things like the Educational Level of the Respondent’s Father.
      2. By time-invariant effects, we mean the variable has the same effect across time, e.g. the effect of gender on the outcome at time 1 is the same as the effect of gender at time 5.
      3. If either of these assumptions is violated, we need to have explicit measurements of the variables in question and include them in our models. In the case of time-varying effects, we can include things like the interaction of gender with time. We also need explicit measurements of time-invariant variables if they are thought to interact with other variables in the model, e.g. we think the effect of SES differs by race.
2. How much variability is there within subjects?
   1. If subjects change little, or not at all, across time, a fixed effects model may not work very well or even at all. There needs to be within-subject variability in the variables if we are to use subjects as their own controls. If there is little variability within subjects then the standard errors from fixed effects models may be too large to tolerate.
   2. Conversely, random effects models will often have smaller standard errors. But, the trade-off is that their coefficients are more likely to be biased.
3. Do we wish to estimate the effects of variables whose values do not change across time, or do we merely wish to control for them?
   1. With fixed effects models, we do not estimate the effects of variables whose values do not change across time. Rather, we control for them or “partial them out.” This is similar to an experiment with random assignment. We may not measure variables like SES, but whatever effects those variable have are (subject to sampling variability) assumed to be more or less the same across groups because of random assignment.
   2. Random effects models will estimate the effects of time-invariant variables, but the estimates may be biased because we are not controlling for omitted variables.

## Other links

Other useful links for Mixed Effects are:

* <https://web.stanford.edu/class/psych252/section/Mixed_models_tutorial.html> for a simple run through of a number of options
* <http://bbolker.github.io/mixedmodels-misc/glmmFAQ.html> gives a more thorough guide to any other issues you may encounter
* <https://stat.ethz.ch/pipermail/r-sig-mixed-models/2010q2/003921.html> provides an explanation to building models

**Creative commons licencing**

Question mark symbol created by Seymur Allahyarli from Noun Project

Exclamation mark symbol created by Michal Beno from Noun Project

1. See appendix A for more information regarding mix/max/averages within linear and log models [↑](#footnote-ref-1)
2. See [appendix B](#_Appendix_B_-) for information on the two different ways of adstocking [↑](#footnote-ref-2)
3. See <https://arxiv.org/pdf/1406.5823.pdf> and search uncorrelate as well as <https://rpubs.com/Reinhold/22193> and <https://stats.stackexchange.com/questions/57240/how-do-i-interpret-the-correlations-of-fixed-effects-in-my-glmer-output> for more information on correlation of mixed effects [↑](#footnote-ref-3)
4. <https://stats.stackexchange.com/questions/242109/model-failed-to-converge-warning-in-lmer> [↑](#footnote-ref-4)
5. See <https://stats.stackexchange.com/questions/242109/model-failed-to-converge-warning-in-lmer> [↑](#footnote-ref-5)
6. https://stats.stackexchange.com/questions/37647/what-is-the-minimum-recommended-number-of-groups-for-a-random-effects-factor [↑](#footnote-ref-6)
7. Copied from https://www3.nd.edu/~rwilliam/stats3/panel04-fixedvsrandom.pdf [↑](#footnote-ref-7)