ARUP

UNIVERSITY OF LEEDS
Institute for Transport Studies

Department for Transport

Audit of Distributional Weight Analysis

Code Audit technical note

Reference: TTWO0213

Revision 2 | 01 August 2023

Job number

Ove Arup & Partners Limited
8 Fitzroy Street
London
W1T 4BJ
United Kingdom
arup.com

ARUP

Document Verification

Project title	Audit of Distributional Weight Analysis
Document title	Code Audit Technical Note
Job number	
Document ref	
File reference	

Revision	**Date**	**Filename**	20230711 Technical note V001B		
1	18 July 2023	**Description**	Version 1		
			Prepared by	**Checked by**	**Approved by**
		Name	Emily Wade, Kajal Kumar	Thijs Dekker, Richard Batley	Adriana Moreno Pelayo
		Signature	EW, KK	TD, RB	AMP
2	01 August 2023	**Filename**	Version 2		
		Description			
			Prepared by	**Checked by**	**Approved by**
		Name	Emily Wade, Kajal Kumar	Thijs Dekker, Richard Batley	Adriana Moreno Pelayo
		Signature	EW, KK	TD, RB	AMP
		Filename			
		Description			
			Prepared by	**Checked by**	**Approved by**
		Name			
		Signature			

Issue Document Verification with Document ✓

Contents

Tables

Figures

1. Introduction

1.1 Background

Arup and the Institute for Transport Studies at the University of Leeds (ITS Leeds) were commissioned by the Department for Transport (DfT) to review and audit the code and methodology of the distributional weighting calculations provided by DfT (Supplier Info). This Technical Note sets out the approach to the code (and relevant Excel spreadsheet) audit and key findings. This note constitutes the first deliverable of the Audit of Distributional Weight Analysis project (Task 1).

1.2 This technical note

This Technical Note summarises the purpose of the audit, giving a summary on the flow of data through the code, the purpose of the code and the key findings of the audit. Further details and a line-by-line code audit can be found in the accompanying excel file (Detailed code audit.xls).

The structure of this Technical Note is as follows:

- Background
- Approach
- Flow Chart of File Structure
- Audit of STATA code
- Audit of R code

Each section below discusses files audited, key findings, checks and excel files audited.

1.3 Approach

The review of the code and associated spreadsheets was led by Arup with input from ITS Leeds. Before starting the tasks, an inception meeting was held to agree on scope and structure of deliverables. The agreed deliverables from the code review are:

- Technical note detailing process and key findings (this note)
- Separate comments tracker with specific issues highlighted by priority and category

Arup engaged with DfT before beginning the audit to confirm the documentation that was relevant for the review and to get an initial briefing on the purpose of the code including critical sections requiring special attention.

A core part of the review focused on outputs and data processing. The review of the code was done line by line, with clarifications obtained from ITS Leeds throughout the process where necessary. We classified issues by priority (high, medium, low) and category (method, error, best practice) and logged them in a tracker (Detailed code audit.xlsx). The code audit followed the Departments' Aqua book principles and Strength in Numbers framework.

Once the code had been reviewed, Arup had a meeting with ITS Leeds to check synergies between Work Stream 1 and 2 and assess the methodological aspects of the code collaboratively. This enabled us to check if there were any aspects to add to the audit results. Findings from this meeting were added to the code audit.

1.3.1 Limitations of the review

The review has excluded the review of some input data. Specifically, NTS and NRTS data inputs were not reviewed, which were assumed to be correct. Some pre-processing input data for STATA was checked as requested by DfT. All files reviewed are listed within this Technical Note.

1.4 Flow chart of file structure

An overview of the STATA code structure is provided below. The STATA code incorporated a wide range of input data as shown in Figure 1.

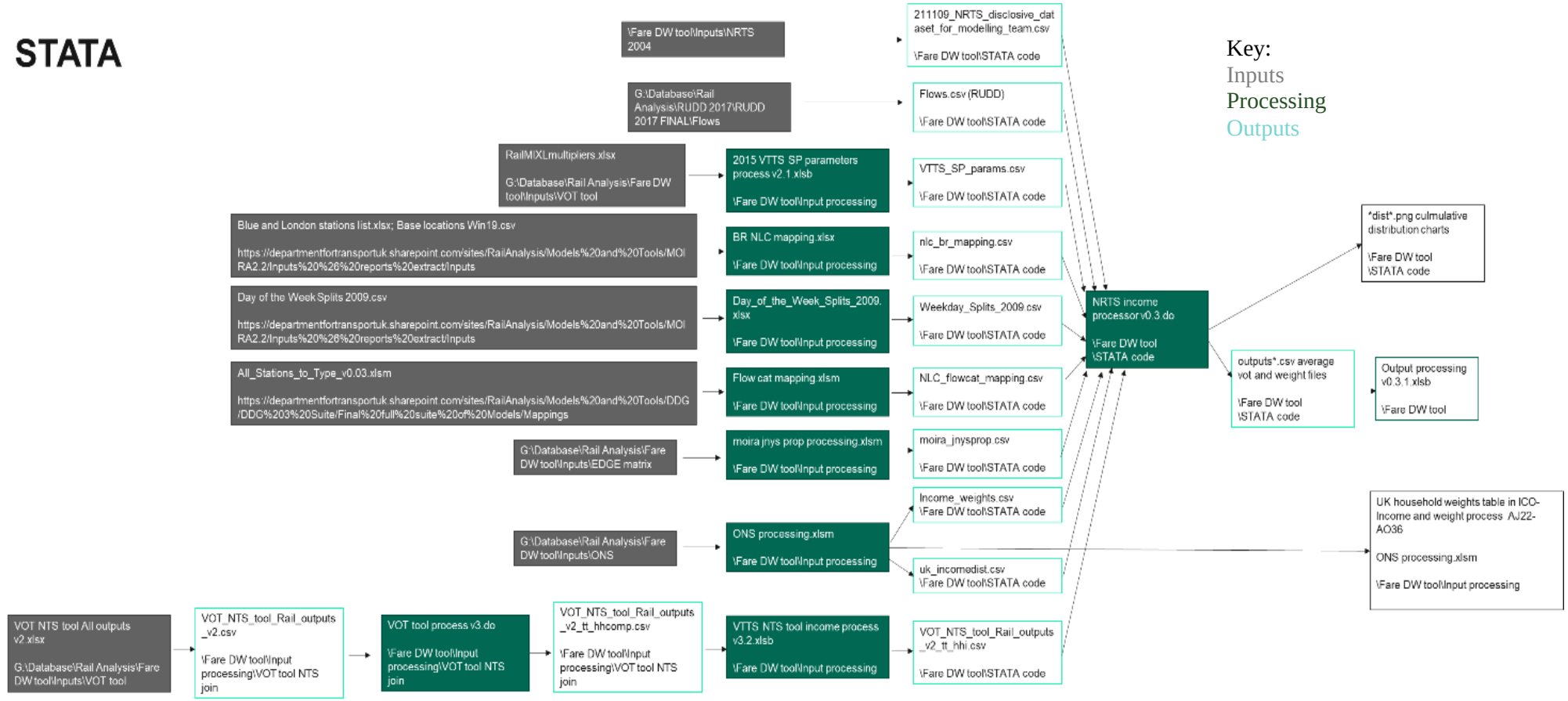

Figure 1: STATA Flow Chart (Source: DfT)

An overview of the R code structure is provided below. The R code incorporated fewer input data files than STATA, with six input data files and four data output files which were then further processed to produce one final excel output file. This is shown in Figure 2.

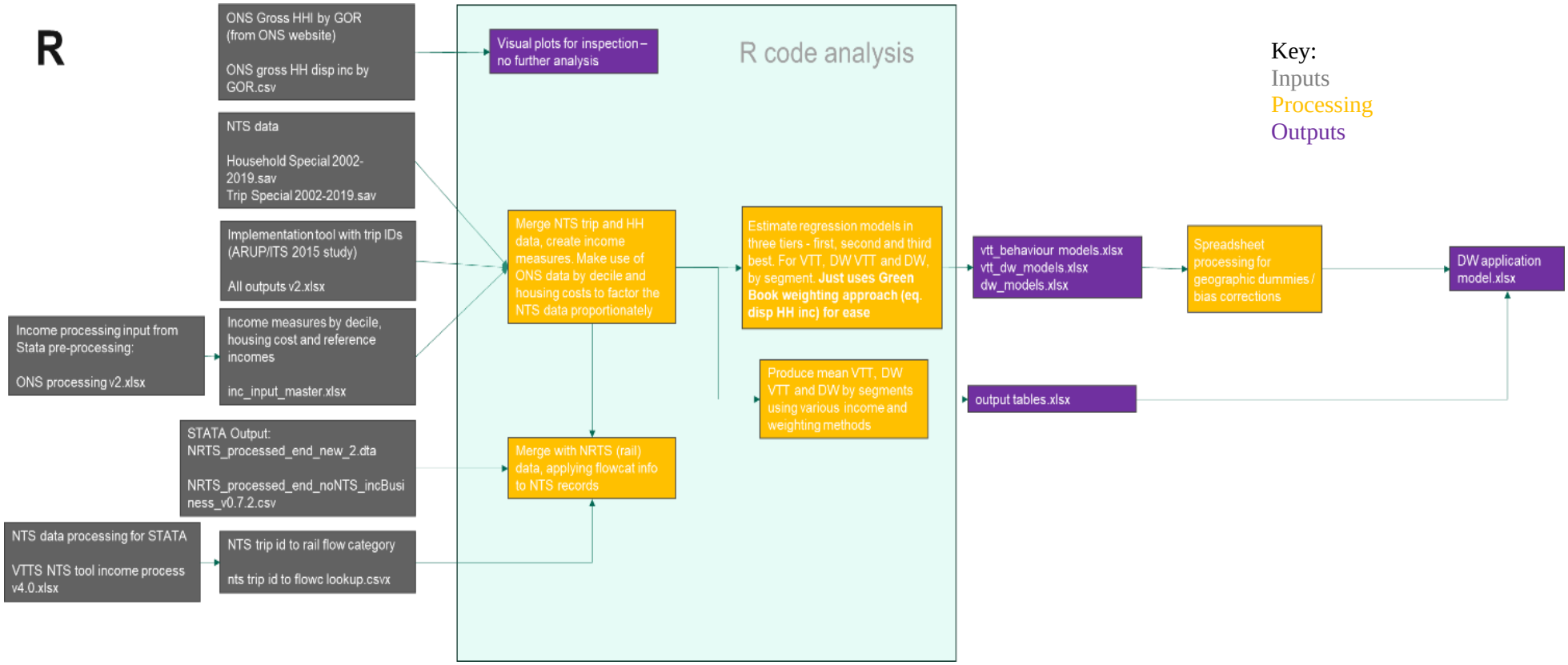

Figure 2: R Flow Chart (Source: DfT)

2. Audit of STATA code

2.1 Summary of files reviewed

We reviewed (in detail) the following files as provided by DfT:

Table 1 Files provided by DfT related to the STATA code

File name	Purpose
TTWO0213_Supp info 1A - Code part 1 - Stata analysis	This file incorporates the code that imports and cleans the NRTS (National Rail Travel Survey) data, combines it with NTS (National Travel survey), RUDD (Rail usage and demand drivers dataset) and MOIRA. It then applies the 2015 VTTS (value of travel time savings) model to the new data to get rail VTTS, related distributional weights (dw) and distributionally weighted VTTS (VTTS_dw) which are used in the R code.
VTTS NTS tool income process v5.0 – HS2.xlsb	This spreadsheet process NTS for appending to the NRTS dataset in STATA.
VOT tool process v3.do	This STATA do file does some more processing of the NTS and NRTS.
ONS processing v2.xlsm	This spreadsheet processes the raw income weights data from the ONS and gives us our income distributions and weights
Moira jnys prop processing v2.xlsm	This spreadsheet processes the MOIRA raw data and calculates journey proportions

2.2 Key findings

Overall, the code is well written and achieves its intended purpose (calculating VTTS, VTTS_dw and dw). The code makes good use of loops and is efficient / non-repetitive, with good commenting and signposting throughout. Some minor errors were identified that can be easily corrected.

The most important suggested improvements are related to readability and defining parameters and formulas more explicitly to make the code easier to follow, should users be unfamiliar with the input data and pre-processing files. We also recommend adding checks for error handling and validation of input data. Other suggested improvements are minor and relate to the order of the code chunks, and descriptive commenting.

2.3 Checks

The following specific checks were performed on the code.

2.3.1 STATA do file

Table 2 Summary of checks performed on STATA code

Item	Comment
Implementation/Functionality	
Does the code do what it's supposed to do?	Yes – the code executes correctly using the input parameters provided in pre-processing excel files, and calculating VTTS, VTTS_dw and dw.

Does the code use the most appropriate libraries and data types?	Yes – the code uses suitable libraries and data types. There are some instances where string datatypes should be in numeric form, these are flagged in the detailed excel file.
Does each function have a single responsibility?	Not applicable – there are no functions within the Stata code. For loops are used, which is makes the code efficient and it is clear what each loop is doing.
Logic errors and bugs	
Are there any obvious logic errors or bugs?	Yes – There are a few questions around methodology and merging of datasets. For example, the way within PTE flowcats are defined is not accurate and missing certain PTE origin destination pairs. Certain flowcats are not classed in core/non core.
Is there anything that would cause unexpected behaviour?	Yes - We noted a few instances where the methodology is questionable. This includes using average fares rather than actual, high number of unmatched values when matching to RUDD, NRTS multiplier calculation being the same for two variables, method for under 16 equivilisation weights, and MOIRA rescaling. Details can be found in the excel file. Other unexpected behaviour might arise from incorrect data types or values in the input data or input parameters. For example, variables that had NA were not destringed, when they should have been.
Error handling and logging (Robustness tests)	
Is the input data validated before it's used?	Somewhat – although the input data and pre-processing files are well structured. The input data is not validated before being used. Some ideas on how to strengthen the data validation are; consider forcing data types during csv read, checking each column contains values in a valid range, check number of columns is correct, and column headers are correct. When importing csvs and merging, it is important to check any duplicates are dropped, and the proportion of matches, to see the accuracy of the merge.
Error handling for incorrect data types?	No – consider adding checks for NRTS_processed.dta catching (and logging) exceptions.
Error handling for invalid values?	Somewhat – checks were done (sometimes) following merges to see all data has been captured and is valid. More could be done to assess the spread of values between different datasets (MOIRA, NRTS, NTS)
Is the output data validated before it is exported?	Somewhat – cumulative distribution charts were created to check the output dataset is valid. Besides this consider checking each column in the final dta, making sure it contains values in a valid range, column headers are correct, and data is in the right format. For the charts, also consider how these might be displayed and if they need to be scaled, or be interactive, or can remain static images.
Readability and accessibility	
Is the code reproducible?	Yes – only a small error in the filename of VOT NTS Tool csv, which causes an error in running (easy to resolve).

Are the names of variables easy to understand?	Somewhat – We would recommend that all data inputs and outputs come with a READ ME that define variable names. There are instances when variable names are improved which is good, but many acronyms exist (BLUE, LAT, ROC). Variables such as 'journeytypeid' or 'originpurpose' are coded as numbers, so it should be clearly stated what each number stands for (this can be done by adding value labels). At one point, distribution weights are referred to as equity weights. This is a little confusing.
Does the order of code chunks make sense? (design)	Yes – the order of the code is sensible. We would suggest moving the main calculations of VTTS_dw, VTTS, and DW, and out sheets of the final csv before the charts. This will make it clear where the data processing ends, and visualisation starts.
Is the code easy to understand? (complexity)	Yes, the code is understandable, well sectioned, and logical. It would be good to see more comments where there are loops to explain the purpose of the loop. When merging datasets, it would be good to explain which parts we are looking to obtain from each dataset, and the purpose of the dataset. When applying formulae such as VTTS_dw, define the parameters clearly in comments (or in the dta), rather than only in the pre-processing files. This will make it easier to relate parameters to the formulas. For others to follow your code it's useful to describe what each block of code does, even though it might seem obvious.
Is there documentation?	Yes – there is an accompanying flow chart. Would be useful to have READ ME file which says how the final dataset is created. What sources are combined and the purpose of each one. Would also be useful to have documentation listing the main formulae used, and parameters defined. This document could also (specifically for the STATA code) detail the mathematics and purpose behind each section. Alternatively, formulae could be explained in comments. For non-technical audiences, having formulae written out (or described in plain English) would help to associate formulas with the calculation in the code. Additional documentation on the 'basic VTTS formula' (from the 14/15 study) is needed. Someone who is not as familiar with this study will not understand that there are 11 mode-purpose combinations (i.e. unique equations) to calculate the VTTS per observation in the NTS data and how the weights are used to arrive at an average. It is also recommended as stated above that it would be beneficial if all data inputs and outputs come with a READ ME that define variable names.
Is the code well commented?	Yes – good comments throughout. More detail is welcome in places, as highlighted above.

2.3.2 ONS processing v2.xlsm

This spreadsheet was set up very clearly, with a clean and easy to follow layout.

Care is needed when hiding rows and columns as this may cause hidden column headings (as it did on one tab). We would also caution against linking too many external files as this slows down the speed and may cause inputs to break. There are also redundant calculations in some tabs, for example summing disposable income values with no clear reason why.

There seems to be an error when calculating nominal values in the median income tab. An accompanying note with formula used to calculate nominal values would be useful. In addition, there is an error with the mortgage values (for some columns) in the housing costs by dec tab.

We have raised some questions on method. It would be beneficial to provide reasons for the calculation of equivalised weight for children under 16 (as also mentioned in the STATA code review).

It is recommended that a READ ME accompanies the cover page with descriptions of how the various tabs link together, what the inputs and outputs are, and descriptions of variable names.

2.3.3 Moira jnys prop processing v2.xlsm

This spreadsheet is set up very clearly, with a clean and user-friendly layout. It is very useful to have the source clearly stated at the top. There were smart ways which automated the flow categories and distance bands. Moreover, sensible robustness checks are there to catch any errors in totals.

We have suggested ways to improve readability such as explaining some variables (Distance band km) and moving the 'Output for STATA' heading lower to coincide with the output table.

It is recommended that a READ ME accompanies the cover page with descriptions of how the various tabs link together, what the inputs and outputs are, and descriptions of variable names.

2.3.4 VTTS NTS tool income process v5.0 – HS2.xlsb

This spreadsheet is set up very clearly, with a clean and user-friendly layout. It is very useful to have the source clearly stated at the top.

We have raised some questions on method. Including why railcards and special passes were excluded (was it to prevent including low incomes, which skew distributional weights to the higher end). As mentioned in the code it would also be good to explain how the weights for children under 16 are calculated, as these are not textbook Green Book weights. Finally, we noticed the region codes for Wales and Scotland are different to the rest, worth checking.

It is recommended that a READ ME accompanies the cover page with descriptions of how the various tabs link together, what the inputs and outputs are, and descriptions of variable names.

2.3.5 VOT tool process v3.do

This is an additional do file which processes and cleans the NTS data further. The code is well structured and logical with good comments throughout.

The only suggestion we make is regarding error handling and robustness tests following a merge. It would be good to tabulate '_merge' to catch any errors and check how well the merge did. This helps to see if the merge worked as we wanted to and ensure that no data is overridden.

2.3.6 Other

Reading .xlsx files and moving between different software packages (STATA and Excel) is not recommended. Non .txt and non .csv files come with a lot of overhead causing error reading this type of data in software like STATA. Data manipulation in Excel and graphing should be feasible in STATA and reduce the risk of errors (and offer more transparency and an auditable trail) because you don't have to check individual cells as opposed to syntax.

3. Audit of R code

3.1 Summary of files reviewed

We reviewed (in detail) the following files as provided by DfT

Table 3 Files provided by DfT related to the R code

File name	Purpose
TTWO0213_Supp info 1B - R analysis.R	This file incorporates the code that explores linear model variables and produces the linear model coefficients and bias for the first best, second best and third best models that are noted in Notes 1 in *TTWO0213_Supp info 2B - DW application model note*.
TTWO0213_Supp info 1C vtt_behaviour_models.xlsx	This file takes the vtt behaviour model outputs from *TTWO0213_Supp info 1B - R analysis.R* and collates the weights from all the same tier models together, flags geography specific bias, produces a final bias correction term and Variable Demand Model (VDM) values.
TTWO0213_Supp info 1D vtt_DW_models.xlsx	This file takes the vtt DW model outputs from *TTWO0213_Supp info 1B - R analysis.R* and collates the weights from all the same tier models together, flags geography specific bias, produces a final bias correction term and VDM values.
TTWO0213_Supp info 1E DW_models.xlsx	This file takes the DW model outputs from *TTWO0213_Supp info 1B - R analysis.R* and collates the weights from all the same tier models together, flags geography specific bias, produces a final bias correction term and VDM values.
TTWO0213_Supp info 1F output tables.xlsx	This file takes various vtt and dw results for different groupings of mode, purpose, distance band and geography. This file also includes cut off values for vtts, vtts_DW_GB and dw_for_costs_GB from *TTWO0213_Supp info 1B - R analysis.R.*
TTWO0213_Supp info 2A - DW application model.xlsx	This file takes various vtt and dw results for different groupings of mode, purpose, distance band and geography. This file also includes cut off values for vtts, vtts_DW_GB and dw_for_costs_GB from *TTWO0213_Supp info 1B - R analysis.R.* This file takes the various outputs from the above spreadsheets and collates them to find the intercepts, coefficients (distributional weights) and minimum and maximum values of each model.

3.2 Key findings

Overall, the code is well written in that it achieves its intended purpose and includes a clear structure and good commenting to provide help to users. The main suggested improvements are related to efficiency / repetitiveness which would help readers more quickly understand the code and reduce the possibility of errors. On a side note, the validation could be more prevalent in the code, with more validation done when reading in data, merging to form *all_modes* (for which the majority of script depends on) and before exporting outputs. Other suggested improvements are minor and relate to the cleanliness of the code alongside with more descriptive comments.

3.3 Checks

The following specific checks were performed on the code.

3.3.1 R code

Table 4 Summary of checks performed on R code

Item	Comment
Implementation/Functionality	
Does the code do what it's supposed to do?	Yes – code executes correctly using the input parameters provided and aligns with the methodology stated in *TTWO0213_Supp info 2B - DW application model note*.
Does the code use the most appropriate libraries and data types?	Yes – the code uses suitable libraries and data types. The use of the data type 'factor' has been used well to ensure that categorical data only takes predefined values. It is however recommended that *purpose* within the data frame *all_modes* also be classed as a data type 'factor'.
Does each function have a single responsibility?	Not Applicable - There are no functions within the code, however it is recommended that functions and looping be used to minimise repeated code.
Logic errors and bugs	
Are there any obvious logic errors or bugs?	No – the code demonstrates that it has interrogated the data and produces outputs that align with *TTWO0213_Supp info 2B - DW application model note*. However, there was one small error noted with a filtering for vtts in one case, as explained in the accompanying excel file Detailed code audit.xlsx.
Is there anything that would cause unexpected behaviour?	Somewhat - It is stated in the code comments that "R^2 values are slightly flawed when using results from dredge in code". It is not explained exactly how. It is recommended that this is explained and that the usage of R^2 is explained.
Error handling, logging, and robustness tests	
Is the input data validated before it's used?	Somewhat - There is little data validation done on data before it is used. Some ideas on how to strengthen the data validation are; consider forcing data types during csv read, checking each column contains values in a valid range, check number of columns is correct, and column headers are correct.
Error handling for incorrect data types?	No – consider adding checks for *all_modes* and catching (and logging) exceptions.
Error handling for invalid values?	Yes – Checks were done that categorical data values / factor levels align between *nrts_data* and joined data, as well as checks regarding *all_modes* data structure which allows the user to check all unique values for certain columns and assess the spread between them which is good. Checks were also done to ensure dummies were not too large, although an explanation as to why they were considered not too large would be beneficial in the comments.
Is the output data validated before it is exported?	No – consider checking each column contains values in a valid range. Check column headers and right format.

Readability and accessibility	
Is the code reproducible?	Yes - code executes correctly using the input parameters provided.
Are the names of variables easy to understand?	Somewhat – We recommend that all data inputs and outputs come with a READ ME that define variable names and acronyms.
Does the order of code chunks make sense? (design)	Yes – Well organised and in a logical order. The only code chunk that does not seem well placed is inputs checks, which is recommended to occur after reading in the input data. The superseded, no longer relevant code often interferes with the order of the code chunks, and it is therefore recommended that the code be cleaned of irrelevant and superseded code.
Is the code easy to understand? (complexity)	Somewhat - The code is well commented, sectioned and logical helping the code to be understandable. However, more comments with more detailed descriptions (rather than just using comments to title sections) is recommended. It's useful to describe what each block of code does, even though it might seem obvious. When merging datasets, it would be good to explain which parts we are looking to obtain from each dataset, and the purpose of the dataset. The superseded, no longer relevant code detracts from the readability of the code. It is recommended that the code be cleaned of irrelevant and superseded code. There is a lot of repetition of code, which makes the code difficult to read due to the length of the full code. It is recommended that the repetition of code is minimised using functions and loops.
Is there documentation?	Yes – there is an accompanying flow chart and a report detailing the mathematics behind the code *TTWO0213_Supp info 2B - DW application model note*. It would be useful to have READ ME file (specifically for the R code) which discusses the formulae used (i.e. *vtt_DW_cost*), methodology behind each code chapter and the process of assessing variables to include in the linear regression models i.e. how and why assessing variables using this method and what you are looking for and decisions made based on the results of the assessment. It is also recommended as stated above that it would be beneficial if all data inputs and outputs come with a READ ME that define variable names.
Is the code well commented?	Yes – good comments throughout. More detail is welcome at places, as highlighted above.

3.3.2 TTWO0213_Supp info 1C-1E model files.xlsx

The majority of the spreadsheet was well formulated and clear. It is recommended that for all mode types that weight is found based on mode, purpose, and geography rather than referencing a specific cell in case the order is different in one sheet to the other.

A potential error with a cell not matching the formula in other cells in column was noticed, if this is not an error there should be a note explaining why.

It is recommended that a READ ME accompanies the outputs with descriptions of variable names and acronyms.

3.3.3 TTWO0213_Supp info 1F output tables.xlsx

It is recommended that a READ ME accompanies the outputs with descriptions of variable names and acronyms.

3.3.4 TTWO0213_Supp info 2A - DW application model.xlsx

The spreadsheet was set up very clearly, with a clean and easy to follow layout including sources for where data is pulled from. However, it is recommended that the spreadsheet be cleaned of tabs not used and cells filled in where it currently states “user inputs”.

Small errors were noted with graph references which should be amended.

It was also noted that some data did not align with the stated location of where the data was pulled from, where all data is pulled from should be noted (especially in the case of multiple data sources).

It is recommended that a READ ME accompanies the cover page with descriptions of how the various tabs link together, and what the inputs and outputs are.

It is also recommended that certain cell referencing is reorganised for easier checking/tracking.

It is recommended that a READ ME accompanies the outputs with descriptions of variable names and acronyms.

3.3.5 Other

Reading .xlsx files and moving between different software packages (R, STATA, and Excel) is not recommended. Non .txt and non .csv files come with a lot of overhead causing error reading this type of data in software like R and STATA. Data manipulation in Excel and graphing should be feasible in R and STATA and reduce the risk of errors (and offer more transparency and an auditable trail) because you don’t have to check individual cells as opposed to syntax.