Modeling is Dead; Long Live Modeling: Regime change in model construction

Abstract

This paper examines models submitted to the International Conference of the System Dynamics Society from 2009 to 2017 via the 'Supporting' links on the conference proceedings website. It compares the 2009-2016 models to the models submitted to the 2017 International Conference of the System Dynamics Society, when a sea-level change in acceptance criteria was applied. The results of applying objective and subjective criteria to model construction quality are presented. A graded approach is then proposed to improve system dynamics model construction quality and recommendations are given for further research and quality initiatives.

The system dynamics methodology life cycle has benefited from a considerable amount of work in model[1] formulation, group collaboration, and model testing. However, there is a missing stage in the modeling process, the stage that corresponds to model construction. System dynamics modelers report on models they have built, but readers and reviewers know little of the model construction process and consequently model quality[2]. It is *assumed* that the model is well constructed. It is appropriate and due diligence to improve the quality and therefore the usefulness of models. Therefore, the field requires a measure, or measures, of the quality of the model construction process.

Introduction

The system dynamics methodology life cycle has benefited from a considerable amount of work in model formulation, group collaboration, and model testing. It has not been as fortunate when it comes to model construction or modeling mechanics.

In the absence of full reporting on the system dynamic modeling process, supporting documents, including models, represent the result of the process and are probably the basis for many of the results and conclusions in the presented papers. An estimate of the quality of the model construction process are then made against the supporting models; proverbially, 'The proof is in the pudding'.

Kubanek (2002) speaks of stages of system dynamics mastery (excerpted here).

> **Stage 1: Novice**
>
> They have no experience of SD. They must be taught context-free rules to guide actions, which is extremely limited and inflexible

[1] Model, the word, can have many meanings. Here we mean the actual software file and all the variables in it. Adding an interface to a model results in an application; model + interface = application. Here I am concerned only with models.

[2] Of course, model testing must also be performed.

Stage 2: Advanced Beginner

They have demonstrated marginally acceptable performance, have coped with some real situations so that they realize that a particular method or knowledge application is situation dependent. The advanced beginner should begin to formulate guiding principles [guidelines] that dictate actions based upon experience.

Stage 3: Competent

They have been practicing SD for several years and thus have experience to see their actions in terms of long-term impacts and are becoming conscious of the strengths and weaknesses. The competent SD modeler lacks the speed and flexibility of the proficient SD modeler but does have a feeling of mastery and the ability to cope with contingencies coming from the unexpected. There is a conscious, deliberate planning required to achieve efficiency and organized work.

Stage 4: Practitioner

They perceive situations as wholes rather than as unique situations full of details to be understand alone and are no longer dominated by guidelines. The practitioner still uses guiding principles as guides, but a deep understanding of the situation is required before a maxim is used. Maxims reflect what would appear to the competent as unintelligible nuances of the situation as they mean difference things at different times.

Stage 5: Expert

They no longer rely on any simple analytic principle [rule, guideline, maxim] as they can connect their understanding of the situation directly – almost intuitively – to the appropriate action.

And yet Kubanek only hints at (guidelines, principles, mastery) but does not specifically address model construction prowess.

The International Conference of the System Dynamics Society makes public the work of students and practitioners of the system dynamics methodology. The Society website makes models freely available via the conference paper listing as supporting documents. These documents can be downloaded and examined (assuming one has the appropriate software). The quality of those models reflects on the System Dynamics Society and the system dynamics methodology. ***<u>The quality of the models reflects poorly on both.</u>***

The concept of 'goodness' when applied to models is difficult to interpret. Goodness is not equivalent to 'truth'. Are there both subjective and objective criteria that can be applied to model quality? Is there a genuine certitude to quality or is quality relativistic? Is this a matter of taste[3] or a matter of truth? Adler (1984) tells us:

***De gustibus non disputandem est* – About matters of taste there is no point in arguing.**

***De veritate disputandum est* – About matter of truth, dispute is fruitful.**

[3] In a discussion with a respected system dynamics scholar a dispute occurred as to whether or not model constants should be in uppercase or lowercase characters. The argument for uppercase pointed to highlighting exogenous influences, the argument for lowercase pointed to minimizing exogenous influences.

Literature review

As early as 1975 Forrester (1975) specified the difference between the model and the modeling process. Many authors have described methods for improving the modeling process including group facilitation and group model building. Martinez-Moyano and Richardson (2013) report on a literature and expert survey that encompasses the six stages of the system dynamics modeling process. They present best practices for each of the stages. The paper might be more appropriately titled "Best practices in using system dynamics for problem analysis". Although there are 72 best practices noted, none address what I term 'the missing seventh stage', model construction.

There are some recommendations for the modeling process including recent guidelines for learning and teaching systems dynamics. Keating (1999) provides 40 pages of recommendations to prepare a model for critique (and likewise to critique a model). Calls for improving the modeling process can be found in Lyneis and Hines (2003), Sterman (2001), Zagonel and Corbet (2006), Meadows and Robinson (1985). More recently Fiddaman (2009, 2010) has suggestions for conference papers. All address issues with model testing and model quality in general plus presentation of model results. None address model construction.

What is a model for our purpose?

A system dynamics model is an organized collection of variables that serve some purpose. Ideally the model executes and passes the tests appropriate to a system dynamics model and gains the confidence of the constructor and client, e.g. student and professor.

Model contents (variables) can be disaggregated as seen in Figure 1.

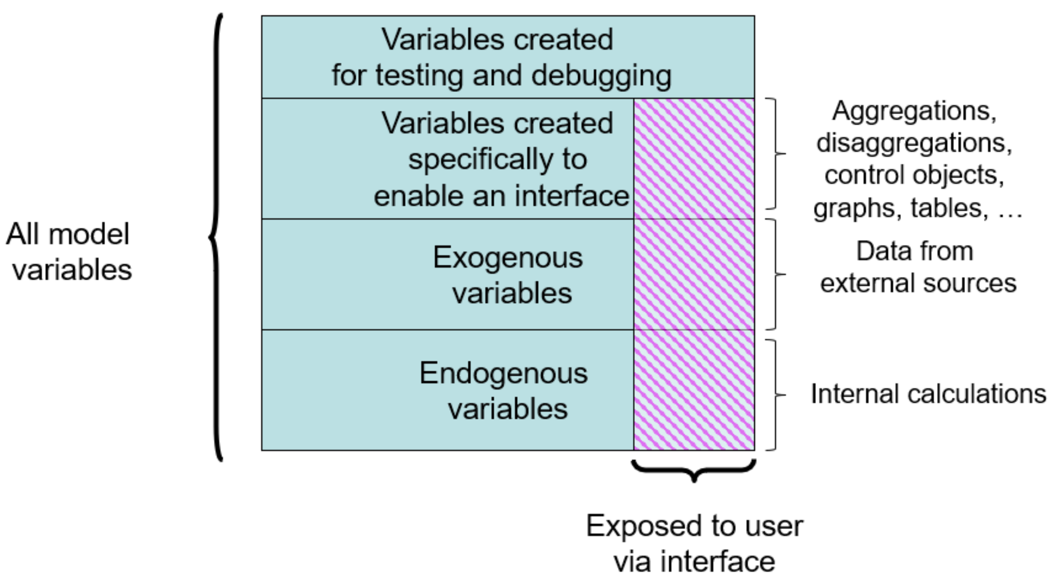


Figure 1 Model variables categorized

Methodology

The list of all papers from the International Conference of the System Dynamics Society years 2009-2017 were loaded into a file and parsed for the word 'Supporting'. Each of the supporting files were downloaded and individually examined. Some of the files contained material other than a model. Figure 2 shows the overall distribution all file types included in supporting documents during a five-year sample period (2009-2013). A similar list of supporting file types exists for the years 2014-2017 with the addition of R and Java Script files. Tables 1 and 2 summarize the supporting documents in the 5-year sample period. Models using Vensim, and Studio[4] were selected and examined using objective and subjective criteria[5].

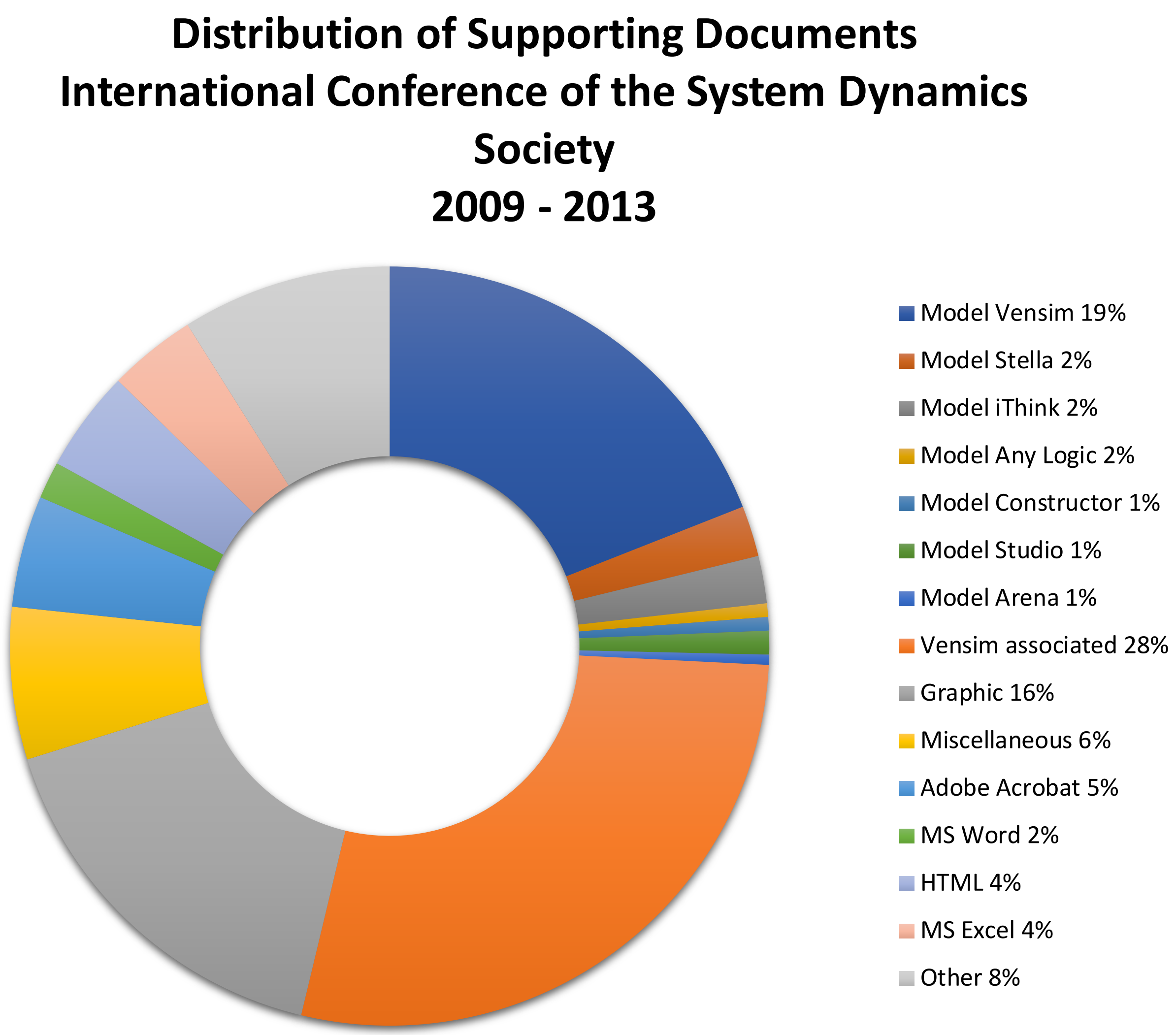


Figure 2 ICSDS supporting documents distribution

[4] Chosen because of expertise in and ownership of these two software packages. The majority of the models are in Vensim format.
[5] Please refer to Appendix A for more criteria.

Each model was opened and the following data[6] were put into an analysis file for each model:

a. Number of levels
b. Number of auxiliaries
c. Number of constants
d. Number of unit errors
e. Number of modules[7]

Tables 1 and 2 further decompose the contents of the supporting files[8].

Table 1 Supporting document categorization: Models

Category\Year	2009	2010	2011	2012	2013	Total
Model Vensim	32	10	21	47	22	132
Model Stella	3	0	3	0	9	15
Model iThink	2	0	8	2	2	14
Model Any Logic	0	1	0	2	1	4
Model Constructor	0	1	3	0	0	4
Model Studio	0	0	3	3	1	7
Model Arena	3	0	0	0	0	3
Total Models	40	12	38	54	35	179

Table 2 Supporting document categorization: Other submitted supporting files

Category/Year	2009	2010	2011	2012	2013	Total
Vensim associated	57	1	41	71	24	194
Graphic	1	9	1	66	37	114
Miscellaneous	16	2	6	14	7	45
Adobe Acrobat	14	1	9	4	5	33
MS Word	0	0	3	6	2	11
HTML	5	6	0	0	19	30
MS Excel	1	1	11	11	2	26
Text	5	0	4	4	5	18
SAS	0	0	0	13	0	13
Email file	8	0	0	0	0	8
MS Access	6	0	0	0	1	7
Compressed	0	0	0	0	3	3

[6] There are many objective and subjective criteria that can be applied in measuring model quality. Please see references Keating (1999) and Malczynski (2011), Lai and Wahba (2001).
[7] A module is a portion of a model separated on to a different View (Vensim) or Constructor Diagram (Studio).
[8] One recommendation is to expand the naming of attachments to conference papers. A possible rubric might be: Model, Data, Poster, Document, Other.

Category/Year	2009	2010	2011	2012	2013	Total
Net Logo	1	0	2	0	0	3
PowerPoint	0	1	1	1	0	3
DLL	0	0	0	2	0	2
Unknown	0	0	1	0	1	2
C program	0	0	0	1	0	1
Video	0	0	0	0	1	1
XML	0	0	0	0	1	1
Total Associated	114	21	79	193	108	511

The characteristics selected were selected as much for expediency as for critique. Unit errors were included because they can provide a quick way to subjectively[9] judge model quality.

> "To the degree that a model passes tests that it is 'sound, defensible and well grounded' it has that degree of validity and, hence, of being good enough for its purpose. If no tests were passed, the model would be completely invalid and hence useless. A model might, however, pass many tests but fail one that is absolutely essential, such as, in system dynamics, dimensional consistency. Such a model would be invalid as one would not know how much confidence could be placed in its outputs." [Coyle and Exelby, 2000]

Of the models examined 35 had sufficient problems that they could not be simulated; 294 were of sufficient quality to run. Ratios were calculated:

1. Variables / Module
2. Variables / Unit error
3. Unit errors / Variable
4. (Levels + Constants) / Total variables
5. Levels / Total variables

The number of models per year is 33, 13, 19, 40, 40, 40, 39, 29, 35 for 2009-2017 respectively.

In addition, due to the wide range in total variables (8 to 34437)[10] the Log_{10} of total variables was calculated. It is important to note what criteria were not applied to the analysis of these models. Namely:

a. Naming (e.g. capitalization scheme, understandable)
b. Layout (e.g. crossing lines, modularization)
c. Embedded constants (e.g. embedded initial values, add-factoring)

---

[99] It is possible to have a 'correct' model with unit errors, e.g. fractions without units. However, evaluation of a great degree of model quality can be rapidly accomplished by a units (dimensionality) check.

[10] Vensim counts each element in an array as a separate variable; overestimating the number of variables. Dynamics before detail!

d. Use of color (e.g. lack of legend, readability)
e. Documentation (e.g. data sources, complex equations)

A cursory view of some of these last characteristics in submitted models shows considerable poor quality in almost all of them[11]. A pessimistic conclusion might be that no model, other than very small models, models much smaller than the average number of variables in the models sampled (71.8 levels, 313.9 auxiliaries, 47.9 constants) can be examined for model construction quality in a tractable manner and period of time. As a model passes each stage of construction quality the time required to test it increases (Wakeland and Hoarfrost 2005[12]). In the extreme, no model can be verified but confidence in a model can be raised.

Results

Let's examine some descriptive statistics of the raw characteristics. As noted earlier, 2017 is examined separately because of the significant change in acceptance criteria for conference papers as per earlier years.

Table 3 shows the maximum, minimum, and average for the collected values for all models 2009-2017. Table 4 shows 2009-2016 and Table 5 shows 2017.

Table 3 Descriptive statistics of all models 2009-2017

	Levels	Auxiliaries	Constants	Unit Errors	Modules
Maximum	5005	30006	972	201	54
Minimum	0	3	3	0	1
Average	71.8	313.9	47.9	17.1	3.3

Table 4 Descriptive statistics of all models 2009-2016

	Levels	Auxiliaries	Constants	Unit Errors	Modules
Maximum	5005	30006	647	201	38
Minimum	0	3	3	0	1
Average	65.8	266.0	40.9	18.0	3.0

Table 5 Descriptive statistics of all models 2017

	Levels	Auxiliaries	Constants	Unit Errors	Modules
Maximum	2545	10404	972	87	54
Minimum	0	3	7	0	1
Average	117.9	675.8	99.4	9.0	6.2

[11] There is little, if any, indication in the model files of how the model was built, how long it took, who were the authors, how many persons participated, what was its purpose, etc.
[12] Note Wakeland and Hoarfrost performed no model construction quality tests.

Size matters or does it? Size may be a characteristic of the model scope. It may also indicate the runaway nature of detail before dynamics. As models increase in size they become harder to manage and maintain.

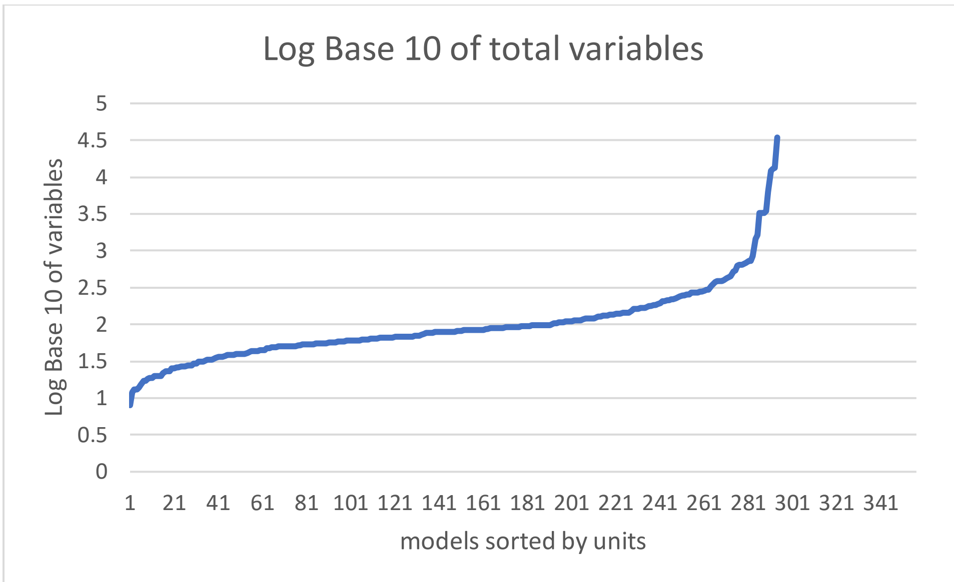


Figure 3 Models sorted by Log10 size

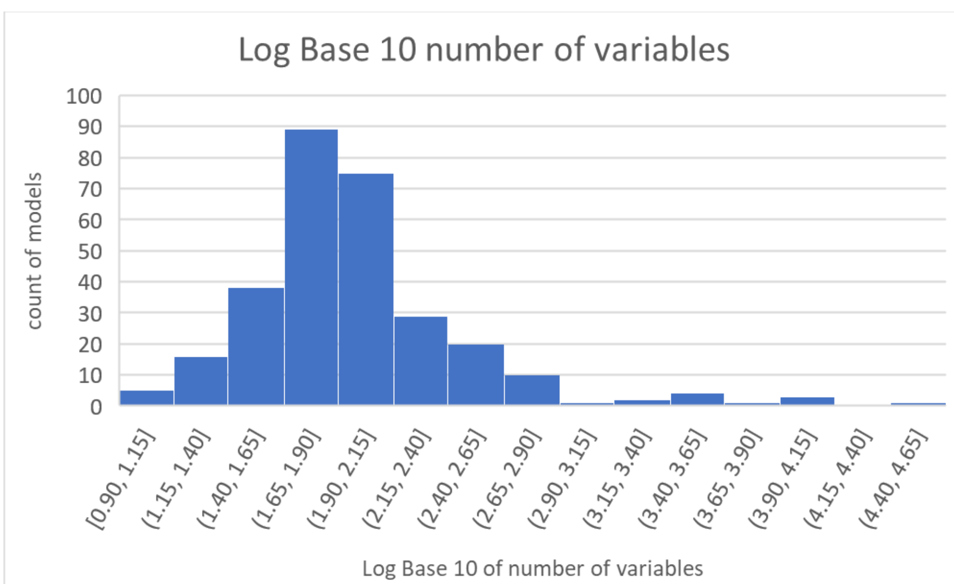


Figure 4 Histogram of model size Log10 of total variables

The ratio variable description statistics are likewise partitioned into all, 2009-2016, and 2017.

Table 6 Descriptive statistics of ratios for all models 2009-2017

	Variables per Module	Variables per Unit errors	Errors per Variable	Total variables	(Level + Constant) / Total	Level / Total
Maximum	13029	3226	1	34437	77%	38%
Minimum	4.1	1	0	8	13%	0%
Average	192.9	45.2	0.2	440.7	45%	14%

Table 7 Descriptive statistics of ratios for all models 2009-2016

	Variables per Module	Variables per Unit errors	Errors per Variable	Total variables	(Level + Constant) / Total	Level / Total
Maximum	13029	3226	1	34437	73%	38%
Minimum	4.1	1	0	8	13%	0%
Average	202.3	44.9	0.2	378.9	45%	14%

Table 8 Descriptive statistics of ratios for all models 2017

	Variables per Module	Variables per Unit errors	Errors per Variable	Total variables	(Level + Constant) / Total	Level / Total
Maximum	1625	494.0	0.9	13337	77%	30%
Minimum	11.8	1.1	0.0	13	15%	0%
Average	127.7	50.1	0.1	907.7	46%	12%

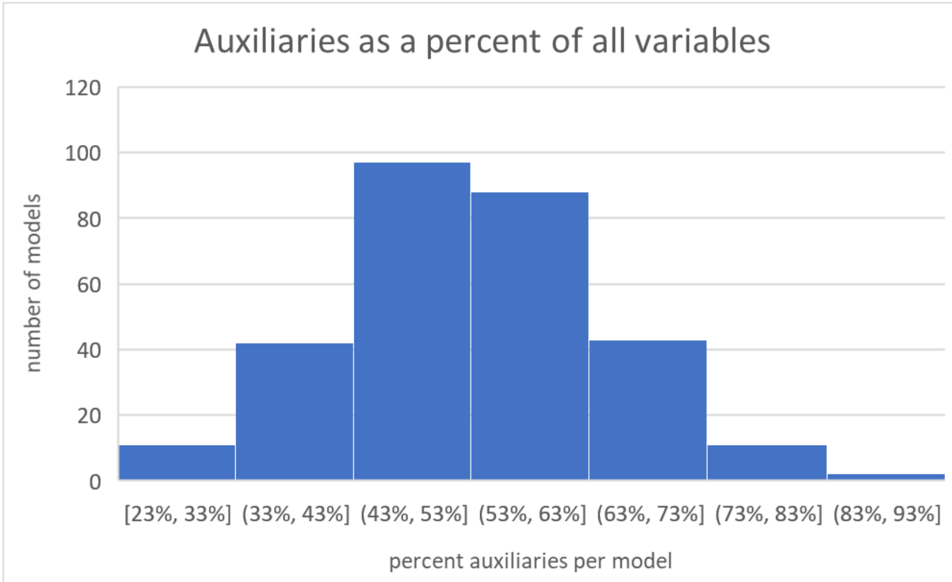


Figure 5 Auxiliaries as a percent of all variables 2009-2017

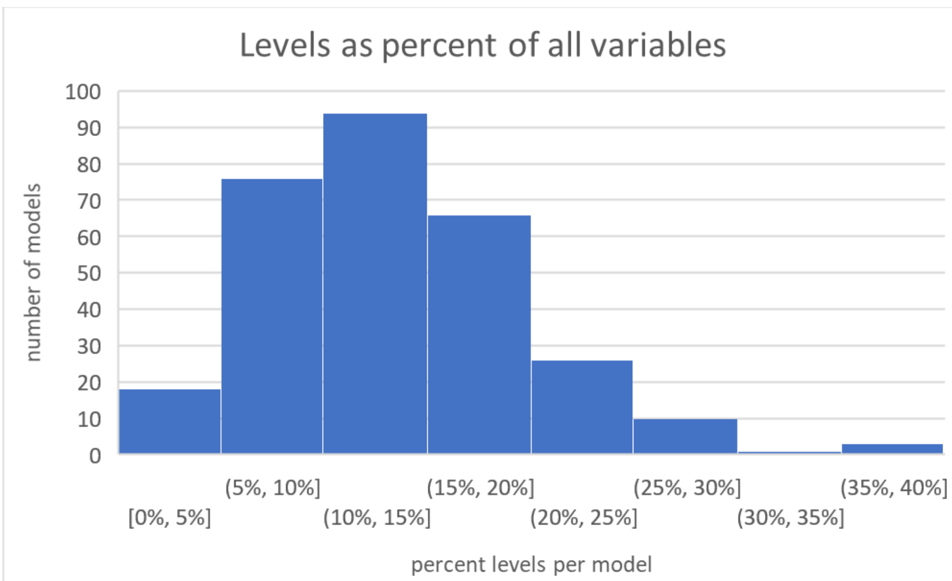


Figure 6 Levels as a percent of all variables 2009-2017

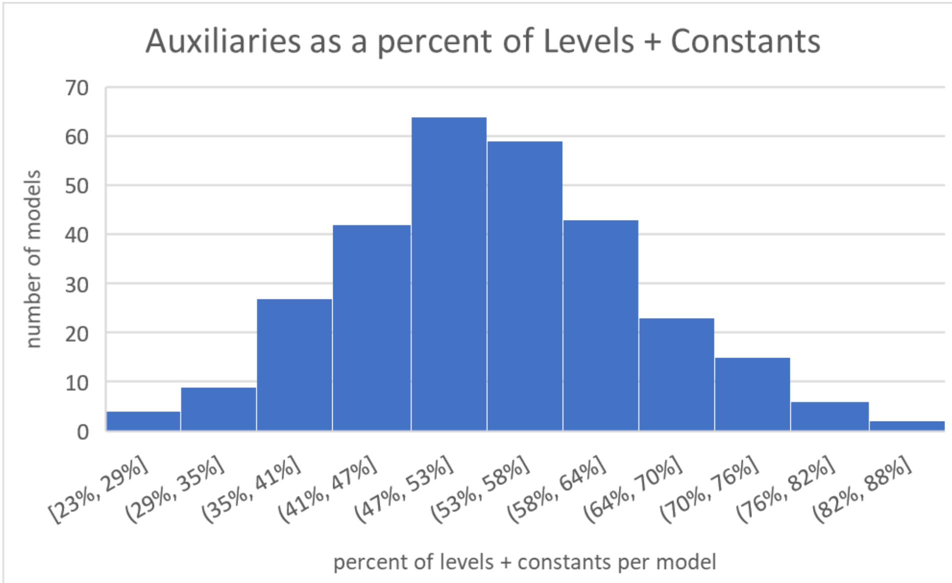


Figure 7 Auxiliaries as a percent of Levels + Constants

Perhaps as models increase in size the number of unit errors increases. This has not been found to be the case. Three figures (Figures 8, 9, and 10) show errors per variable, errors per variable for models with less than 1000 variables, and errors per variable for models with less than 100 variables.[13]

[13] Note that the eye is drawn to the diagonal. The diagonal is the upper error limit, i.e. every model variable having a unit (dimensionality) error.

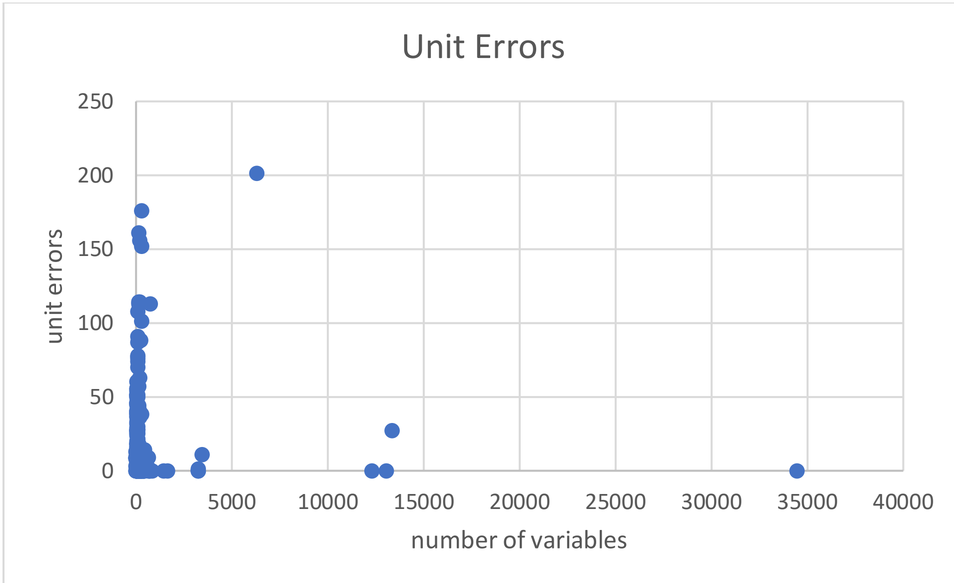


Figure 8 Unit errors by model size (errors on Y axis)

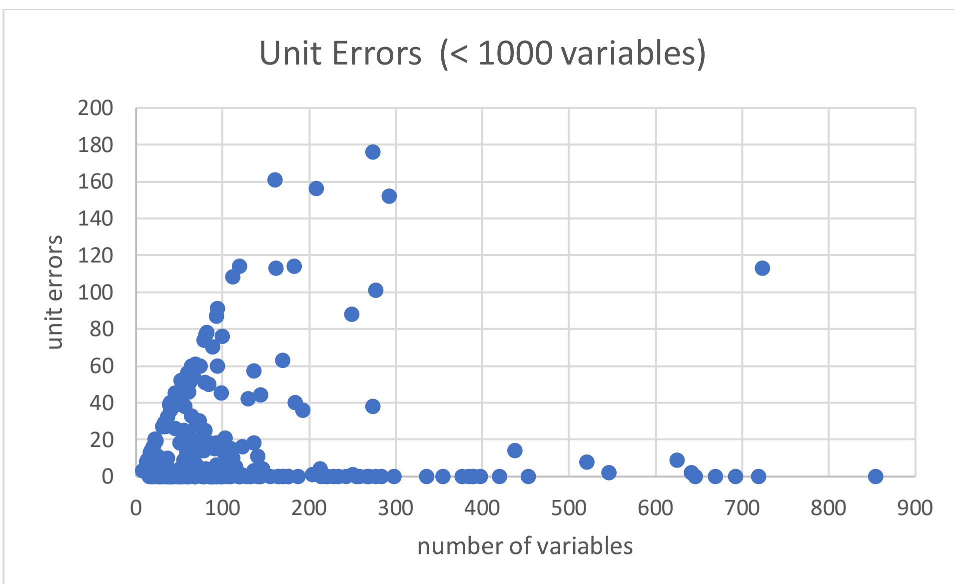


Figure 9 Unit errors by model size for models of less than 1000 variables (errors on Y axis)

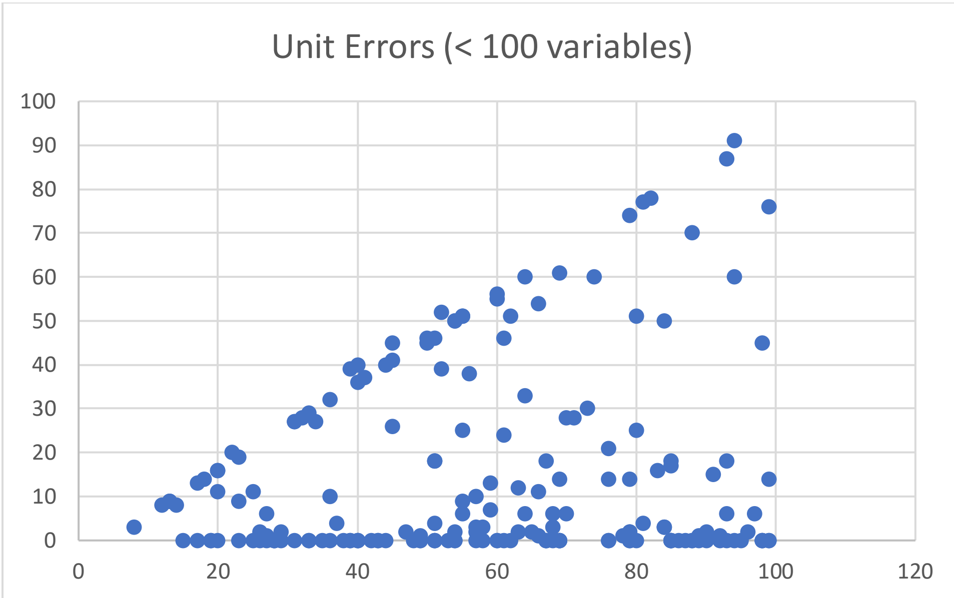


Figure 10 Unit errors by model size for models of less than 100 variables (errors on Y axis)

The diagonal shows the upper limit of unit errors, that is if all model variables have unit errors. Notice the wide distribution and seeming lack of correlation between model size and numbers of errors.

Using the number of levels, auxiliaries, and constants does not provide a clear distinction between the models submitted 2009-2016 and those submitted in 2017.

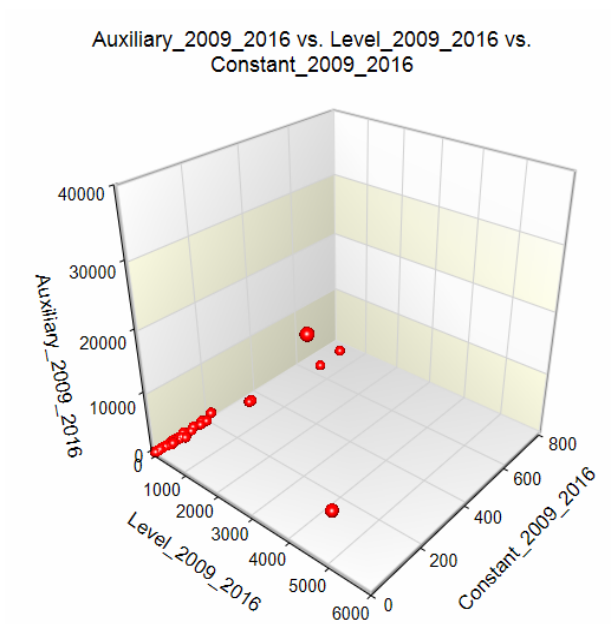


Figure 31 Level vs Auxiliary vs Constant (skewed by counting elements in an array as separate levels)

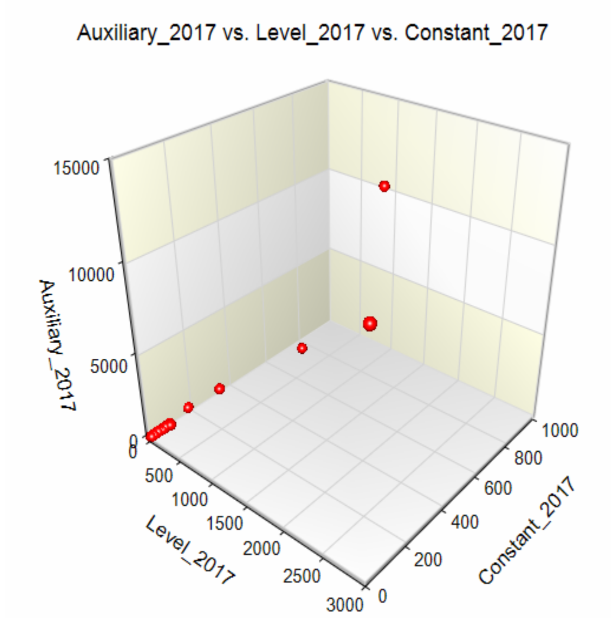


Figure 42 Level vs Auxiliary vs Constant (skewed by counting elements in an array as separate levels)

Analysis of variance and t-tests were performed on variables to determine if the is a statistical difference between data from different conference years. Selected results are in Appendix B. There is a difference between the years although there are pairs of years that do not reject the hypothesis that the means of variables are equal. Much more analysis is required. Even more importantly, we might ask the question, “Can a few variables describe the variation in model quality?” Perhaps this can be accomplished using factor analysis. Finally, further research is required to determine graded quality. For example, unlike dimensionality which is a binary (all variables have units or not), are there other subjective scalable measures of construction quality.

Discussion

Many years of experience with modeling tools have taught us that a bit of prevention is a great cure to model complexity. Some would say that best practice guides tend to stifle creativity. I argue that it liberates creativity in that it simplifies mundane modeling tasks, makes models done by other modelers more accessible, and prolongs the useful life of models. Best practices are a mixture of the pedagogic and the practical. These recommendations have their foundations in software engineering[14] (McConnell 2004) and the system dynamics methodology.

1. Some objective questions concerning model quality

    a. Are there undefined or problematic variables?
    b. Does every variable have a well-defined unit assigned to it?

[14] See Osgood and Tian 2012 for a broader perspective and practices we have not adopted.

c. Does the model follow 'standard' naming conventions?
d. Does the model contain embedded constants?
e. Is there unit consistency between linked variables.
f. Are stocks initialized internally with other than non-zero values?
f. Is every variable used by another variable? If not are simply reporting values?
g. Do variables exist in an equation listing and not on any of the model screens?

2. Some subjective questions concerning model quality

   a. Are variable names well-constructed?
   b. Do modules exist, i.e. is the model partitioned into sub-models?
   c. Is the layout of Constructor diagrams (sheets) instructive?
   d. Does documentation exist?
   e. Do arrays follow good practice guidelines?

Some of these tests can be performed though observation, these are non-destructive. Other destructive tests, tests that require a modification of the model, can be performed to gain a better understanding of the model logic and its construction quality.

Coincidently, the analysis of other models can lead to and has led to best practices for model construction. Much of what we criticize can be constructively used to guide better model building.

Naming

Most software has the capability of permitting long variable names. In fact, several hundred characters are not a problem. These are the styles and recommendations for naming:

1. The names of variables should be clear and unambiguous[15].
2. Resist using uncommon abbreviations, e.g. 'fltp' when you mean 'fuel type'.
3. Use acronyms sparsely and only if they are generally accepted in your field, e.g. kWh instead of kilowatt hours.
4. Resist using digits, e.g. 1, when you mean one. Occasionally digits can be used if your client uses them to name objects, e.g. 'Power plant 1'.
5. Resist using special characters, e.g. !, #, $, &, ~, -[16].
6. Constants should be in uppercase, e.g. INITIAL POPULATION.
7. Auxiliaries, including flow rates, should be lowercase, e.g. population growth rate.
8. Stocks should have the first letter of major words in uppercase, e.g. Corn Stock.

Dimensional consistency

Unitizing variables (dimensional consistency) is one of the most controversial topics in model construction and continues to be so. In the recent past this controversy stemmed primarily from the difficulty in applying and managing units within system dynamics software. Additionally, there is a belief that if a model stays below a certain 'small' size, units are not required; this assumes that the

[15] Gone are the days of DYNAMO when names were constrained to be 4-6 characters long.

[16] These characters can be used in names but may cause confusion in equations especially if exporting equations.

modeler is more than capable of managing the syntax and semantics of the model construction without the overhead of unitized variables if the model is ‘small’.

There is no question that every, yes, every model variable should have a unit assigned to it. Perhaps some clarification of that last comment is necessary. Yes, software can be used to build models that do not follow the system dynamics methodology. Those models can be built without units. I am ignoring those models[17]. As a best practice, assign a unit to each and every variable.

> “To the degree that a model passes tests that it is ‘sound, defensible and well grounded’ it has that degree of validity and, hence, of being good enough for its purpose. If no tests were passed, the model would be completely invalid and hence useless. A model might, however, pass many tests but fail one that is absolutely essential, such as, in system dynamics, dimensional consistency. Such a model would be invalid as one would not know how much confidence could be placed in its outputs.” [Coyle and Exelby, 2000]

One unit that raises some controversy is <<dimensionless>>. When a ratio is calculated in a model the units cancel and you seem to have a unit-less variable. Historically DYNAMO used the <<dmnl>> unit to indicate the unit on these variables. Some modelers also use <<fraction>>. Then there are constants that serve as switches for an interface, since all variables must be assigned units we create a custom unit <<dimensionless>> or <<dmnl>>.

Modularity or Partitioning: Art and Science

“Structured Design seeks to conquer the complexity of large systems in two ways: partitioning and hierarchical organization.” [Page-Jones, 1980]. As models get larger and larger (of course that depends on how one measures large) it is a good practice to modularize your model. An easy step is to make a ‘template’ model. The template would contain commonly used units and views or sheets as recommended below. I suggest these:

- A programmer comments sheet/view – for notes, versioning, etc.
- A best practices sheet/view – used as a reminder
- A scratchpad sheet/view – for items that can be throw away or saved but that are not necessary for the model or interface to function
- A debugging sheet/view for model items that don’t work[18]

Coupling

“… one of the crucial points of this low coupling is that no module has to worry about the particular internal construction details of any other” (Page-Jones, 1980). Coupling and cohesion are the attributes of our ability to partition all the variables in a model into modules (also known as views,

[17] During this analysis a model was discovered with no stocks.

[18] Some modelers never close and save a model unless all variables are correct, no problematic or invalid variables. This practice has been promoted by the Agile Techniques movement.

diagrams, or sheets). This ability involves both modeling skill and creativity. There is no algorithm that leads to the lowest coupling and highest cohesion. There are however many post-modularization tests that can be performed to judge our abilities to modularize[19].

Coupling has nothing to say about this grouping. On the other hand, if a model is separated across multiple modules that separation should reflect that the modules are truly different. The aspect of the difference will be a function of the problem domain. As an example, a model that concerns itself with the limitations on biomass production as an input to cellulosic ethanol production might create a module for each of the biomass sources, e.g. crop residue, short rotation woody crops, and municipal wastes. All three biomass sources have the common characteristic of being a biomass feedstock but each has a different enough conversion process to ethanol that at least three modules would be used to model the supply chain and transformation of each. Looking at them a different way we should see that these modules, crop residue, short rotation woody crops, and municipal wastes, would probably not need to communicate with each other. An example can be found in Figure 13[20]. Each colored circle represents a module and each line represents that there are variables, the number on the line, passed between modules (Jacobsen and Bronson 1985). This brings us to the next topic, cohesion.

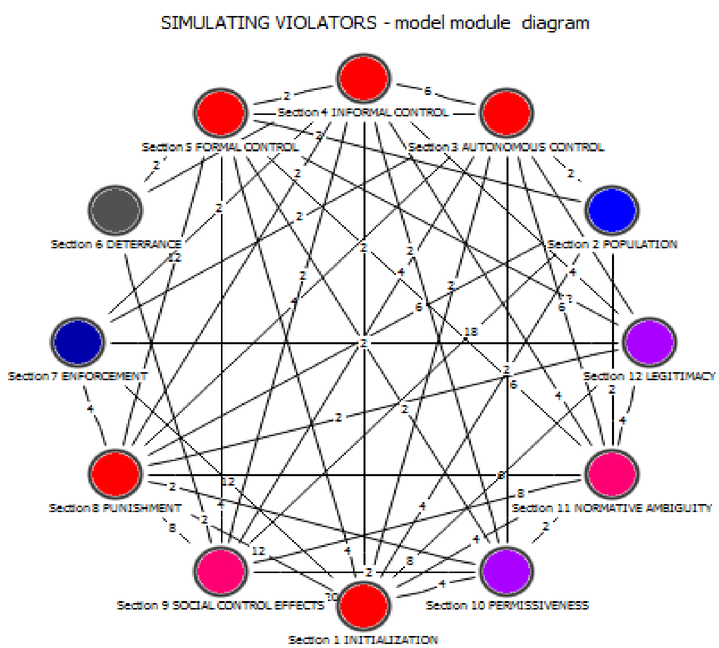


Figure 13 Modules (circles) and coupling (numbers on lines) for Simulation violators (Jocobson and Bronson 1985)

Cohesion

"Cohesion is the measure of the strength of functional association of elements within a module." (Page-Jones, 1980). Models can be quite large (as measured by the number of objects). In order to

[19] Please see Page-Jones for a detailed description of the levels of coupling and cohesion.

[20] This diagram required exporting of the model's variables by module, managing the module to module relationships in a relational database, then using a network graphics program.

maintain sanity and the possibility of re-use we should put objects on separate sheets because we believe that certain variables 'belong together'. Our ability to group variables results in several consequences. As a result, we produce modules that are strong or highly cohesive.

Cohesion has been measured by seven levels of 'goodness'[21]:

Functional
Sequential
Communicational
Procedural
Temporal
Logical
Coincidental

We strive for Functional cohesion. In system dynamics models, Sequential and Temporal cohesion are typically not a problem.

Model naming

Most system dynamics software does not have a built-in configuration management feature at the time of this writing. Experience has shown that saving your model frequently is a good practice. Experience suggests that models be named this way:

Name [Date][a] Tool Tool-version

e.g. FlyswatterSales 20101022 S8FP2SR5.sip

FlyswatterSales 20101022 V603G

Where:

'Name' is an acronym or name for the model content, e.g. FlyswatterSales

'Date' is yyyymmdd

'a' is an optional letter of the alphabet just in case you want to save multiple models on the same day.

The operating system time stamp on the file can serve the same purpose.

'S8FP2SR5' is Studio 8, Feature Pack 2, Service Release 5

'V603' is Vensim 6.03G

We save copies of the model as it gets built. This prevents what software developers call 'bridge burning edits', you find that your current model has a bug and wish you could return to an earlier version but it is

[21] See Page-Jones Chapter 7 Cohesion

gone. Application delivery may use a different convention using numbering that indicates the release number.

Steps forward

There are a number of best practices that can be applied to the model construction process. Many of these steps have earned their merit during the 60-year practice of system dynamics modeling and the equally long tenure of software engineering.

It has been said that wise persons learn from the experience of others. In that respect we do have scores of models that have been somewhat vetted for quality in texts (Sterman, 2001, Bossel 2007), guides (Hines, 1997), and user groups (https://groups.yahoo.com/neo/groups/powersimtools/info, http://www.ventanasystems.co.uk/forum/, etc.). In a sense these models form the object hierarchy, especially Molecules and the Systems Zoo, from which all other models can be constructed.

Suggested future work

Follow-on work for the System Dynamics Society is to build and make available a vetted object hierarchy of quality vetted modules and models in various software formats.

A follow-on study may contact the authors of models that pass demanding objective and subjective quality standards for their recommendations on how to manage model construction much as Martinez-Moyano and Richardson (2013) did for the modeling process.

Begin to fund an SDM Doc tool for other modeling language formats.

Finally, it would be interesting to perform an examination of models submitted in formats other than Vensim and Studio to discover if there is a tool bias with respect to quality..

Conclusion

There are gems in the mix, but it is a lot to sift through.

The quality of models submitted to the system dynamics conference using objective criteria fail to meet the basic characteristics of model quality in taste and truth. Some simple steps can be taken to improve the quality of models at the academic level and by practitioners through the use of simple checklists and good model samples. Doing that will lead to the development of good personal practice. More sophisticated quality checks can be performed using all available and ad hoc tools (for example SDM Doc). As model quality improves researchers inside the field of system dynamics will more readily make use of existing models and researchers outside the field we'll be able to understand the system dynamics modeling that has already occurred. One result will be an improvement in the reputation of the System Dynamics Society and the field of System Dynamics.

Acknowledgement: Thanks to anonymous reviewers who suggested improvements to the graphics and organization of this paper.

References

Adler, M. *Six Great Ideas*. 1984. Collier Books, New York, NY.

Bossel, H. System Zoo Volumes 1, 2, and 3. Books on Demand GmbH, Norderstedt, Germany.

Coyle, G, Exelby, D. 2000. The validation of commercial system dynamics models. *System Dynamics Review* 2000, **17**(4), 357-363.

Fiddaman, T. 2009. Writing an SD conference paper. Retrieved 2018/03/19 http://metasd.com/2009/04/writing-an-sd-conference-paper/

Fiddaman, T 2010. Writing a good system dynamics paper II. Retrieved 2018/03/19, from http://metasd.com/2010/04/writing-a-good-sd-paper-ii/

Hines, J. 2018 Molecules of Structure. Retrieved 2018/05/30 http://vensim.com/modeling-with-molecules-2-02/, Retrieved 2018/05/30 https://groups.yahoo.com/neo/groups/powersimtools/files/Molecules%20of%20Structure/

Jacobsen, C, Bronson, R. 1985. *Simulating Violators*. Operations Research Society of America.

Keating, E. 1999. Issues to consider while developing a System Dynamics model. Keating, E. Issues to consider while developing a System Dynamics model. Kellogg Graduate School of Management Northwestern University.

Kubanek, G. 2002. A System Dynamics Skills Inventory for Self-Taught Practitioners to begin to answer the question: What constitutes System Dynamics Mastery? A report for the Policy Council of the System Dynamics Society.

Lai, D, Wahba, R. 2001. System Dynamics Model Correctness Checklist, MIT D-Memo D-4851.

Lyneis, J, Hines, J. The Standard Method. Retrieved on 2018/05/30 https://ocw.mit.edu/courses/sloan-school-of-management/15-875-applications-of-system-dynamics-spring-2004/projects/stnd_method.pdf

Malczynski, L. 2011. Best Practices in System Dynamics Model Construction with Powersim Studio. SAND2011-4108, Sandia National Laboratories, Unlimited Release.

Malczynski, L. 2014. The missing seventh stage: model construction. 32th International Conference of the System Dynamics Society Proceedings. Delft, The Netherlands.

Martinez-Moyano, IJ, Richardson, GP. 2013. Best practices in system dynamics modeling, *System Dynamics Review* Vol. **29**(2): 102–123

McConnell, S. 2004. *Code Complete*. 2nd Edition, Microsoft Press.

Meadows, D, Robinson, J. 1985. *The Electronic Oracle: Computer Models and Social Decisions*. John Wiley & Sons.

Osgood, N, Tian, Y. 2012. 15 Things System Dynamics can Learn from Software Development. 30th International Conference of the System Dynamics Society Proceedings. St. Gallen, Switzerland.

Page-Jones, M. 1980. *The Practical Guide to Structured Systems Design*. Yourdon Press, New York, NY.

Papachristos, G. 2018. Demos and models from the literature. Retrieved 2018/03/22 https://groups.yahoo.com/neo/groups/powersimtools/files/Demos%20%26%20Models%20from%20literature/

Qudrat-Ullah, H. 2005. STRUCTURAL VALIDATION OF SYSTEM DYNAMICS AND AGENTBASED SIMULATION MODELS, Proceedings 19th European Conference on Modelling and Simulation Yuri Merkuryev, Richard Zobel, Eugène Kerckhoffs © ECMS.

Rahmandad, H. and Sterman John D. 2012. Reporting guidelines for simulation-based research in social sciences. *System Dynamics Review* **28**(4) 396-411.

Wakeland, W, Hoarfrost, M. 2005. The case for thoroughly testing complex system dynamics models. International Conference of the System Dynamics Society, Boston, MA.

Zagonel, AA, Corbet, TF. 2006. Levels of Confidence in System Dynamics Modeling: A Pragmatic Approach to Assessment of Dynamic Models. International Conference of the System Dynamics Society, Nijmegen, The Netherlands.

Appendix A: Group model construction and model reuse criteria

<u>Objective criteria</u>	Description
Variable count	The total number of variables includes exogenous inputs, endogenous model calculations and variables for the interface.
Element count	This represents the number of model values. An arrayed-variable counts as one variable and as many elements as its dimension.
Element/Variable ratio	This ratio is in some sense a measure of model leverage. Variables represent dynamics, elements represent details.
Relative model size	Model size in variables divided by the average model size in variables. This is useful for groups that have an archive of models.
Ranges	A range is a variable's dimension, e.g. a range called 'States' would have 50 elements, one for each state.
Defined units count	Either atomic units or SI units..
Adequate units	A binary decision: all variables have units or not all variables have units.
Modularity	The model is logically divided into relatively self-contained sections if necessary.
Model	Decomposition of a large problem into modules is accomplished with tabs. The number of model tabs signals the degree of model decomposition and re-usability.
Interface	The number of interface elements or views.
Other	Other model decompositions
Stocks	Stocks or levels are the model variables that accumulate material, information, persons, etc.
Auxiliaries/Flows	Auxiliaries are composed of rates (flow into stocks per unit of time) and variables used for any purpose other than stock or rate.
Constants	Constants signal the degree to which the model output is controlled by external variables and not by causal relationships in the model itself.
Data quality: constants have documentation	Are all constants documented?
Modeling conventions	These conventions were developed by modelers with input from many sources. If followed, they improve the understandability and reusability of the model.
Naming	Is a well-defined naming convention used?

Embedded constants	Are there auxiliaries with embedded and undefined constants?
Variable names well defined	Are the variables named using the naming convention?

<u>Subjective criteria – applied as experience in modeling is gained</u>
Organization score* [22]
Sufficient documentation exists: to undertake improvement by original author(s)
Sufficient documentation exists: to reproduce results by non-authors
Percent of constants documented and documented sufficiently
*Subjective score with values Awful, Poor, Good, Very Good and Excellent

[22] A subjective criterion.

Appendix B Statistical output

Two-Sample Test Report

Dataset	Total levels
Group 1 Variable	Level_2009_2016
Group 2 Variable	Level_2017
Difference	(Level_2009_2016) - (Level_2017)

Descriptive Statistics

Variable	Count	Mean	Standard Deviation of Data	Standard Error of Mean	T*	95% LCL of Mean	95% UCL of Mean
Level_2009_2016	259	65.57915	423.8633	26.33758	1.9692	13.71515	117.4432
Level_2017	35	117.9143	450.0281	76.06863	2.0322	-36.67576	272.5043

Descriptive Statistics for the Median

Variable	Count	Median	95% LCL of Median	95% UCL of Median
Level_2009_2016	259	10	8	11
Level_2017	35	11	8	18

Two-Sided Confidence Interval for μ1 - μ2

Variance Assumption	DF	Mean Difference	Standard Deviation	Standard Error	T*	95% C. I. of μ1 - μ2 Lower Limit	Upper Limit
Equal	292	-52.33514	426.9923	76.89708	1.9681	-203.6779	99.00765
Unequal	42.56	-52.33514	618.2114	80.49909	2.0173	-214.7256	110.0553

Equal-Variance T-Test

Alternative Hypothesis	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at α = 0.050?
μ1 - μ2 ≠ 0	-52.33514	76.89708	-0.6806	292	0.49667	No

Aspin-Welch Unequal-Variance T-Test

Alternative Hypothesis	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at α = 0.050?
μ1 - μ2 ≠ 0	-52.33514	80.49909	-0.6501	42.56	0.51910	No

Two-Sample Test Report

Dataset	Total variables
Group 1 Variable	Total_variables_2009_2016
Group 2 Variable	Total_variable_2017
Difference	(Total_variables_2009_2016) - (Total_variable_2017)

Descriptive Statistics

Variable	Count	Mean	Standard Deviation of Data	Standard Error of Mean	T*	95% LCL of Mean	95% UCL of Mean
Total_variables_2009_2016	663.7791	259	377.6139	2338.71	145.3204	1.9692	91.44877
Total_variable_2017	35	907.6572	2989.127	505.2548	2.0322	-119.144	1934.458

Descriptive Statistics for the Median

Variable	Count	Median	95% LCL of Median	95% UCL of Median	
Total_variables_2009_2016		259	79	69	85
Total_variable_2017	35	93	85	146	

Two-Sided Confidence Interval for μ1 - μ2

Variance Assumption	DF	Mean Difference	Standard Deviation	Standard Error	T*	95% C. I. of μ1 - μ2 Lower Limit	Upper Limit
Equal	292	-530.0432	2423.44	436.4375	1.9681	-1389.005	328.9188
Unequal	39.82	-530.0432	3795.319	525.7379	2.0214	-1592.747	532.6607

Equal-Variance T-Test

Alternative Hypothesis	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at α = 0.050?
μ1 - μ2 ≠ 0	-530.0432	436.4375	-1.2145	292	0.22555	No

Aspin-Welch Unequal-Variance T-Test

Alternative Hypothesis	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at α = 0.050?
μ1 - μ2 ≠ 0	-530.0432	525.7379	-1.0082	39.82	0.31945	No

Two-Sample Test Report

Dataset	Unit errors per variable
Group 1 Variable	Unit_error_per_variable_2009_2016
Group 2 Variable	Unit_error_per_variable_2017
Difference	(Unit_error_per_variable_2009_2016) - (Unit_error_per_variable_2017)

Descriptive Statistics

Variable	Count	Mean	Standard Deviation of Data	Standard Error of Mean	T*	95% LCL of Mean	95% UCL of Mean
Unit_error_per_variable_2009_2016	0.1914076	0.2744173	259	0.2329125	0.3392022	0.021077	1.9692
Unit_error_per_variable_2017	0.04172354	35	0.1350702 0.2284168	0.271742	0.04593278		2.0322

Descriptive Statistics for the Median

Variable	Count	Median	95% LCL of Median	95% UCL of Median		
Unit_error_per_variable_2009_2016			259	0.01877934	0.003120125	0.04938272
Unit_error_per_variable_2017		35	0	0	0.02739726	

Two-Sided Confidence Interval for μ1 - μ2

Variance Assumption	DF	Mean Difference	Standard Deviation	Standard Error	T*	95% C. I. of μ1 - μ2 Lower Limit	95% C. I. of μ1 - μ2 Upper Limit
Equal	292	0.09784228	0.332053	0.05979945	1.9681	-0.0198503	0.2155349
Unequal	49.54	0.09784228	0.4346284	0.05053771	2.0090	-0.003689268	0.1993738

Equal-Variance T-Test

Alternative Hypothesis	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at α = 0.050?
μ1 - μ2 ≠ 0	0.09784228	0.05979945	1.6362	292	0.10288	No

Aspin-Welch Unequal-Variance T-Test

Alternative Hypothesis	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at α = 0.050?
μ1 - μ2 ≠ 0	0.09784228	0.05053771	1.9360	49.54	0.05858	No

One-Way Analysis of Variance Report

Dataset	Untitled
Response	C1,C2,C3,C4,C5,C6,C7,C8,C9

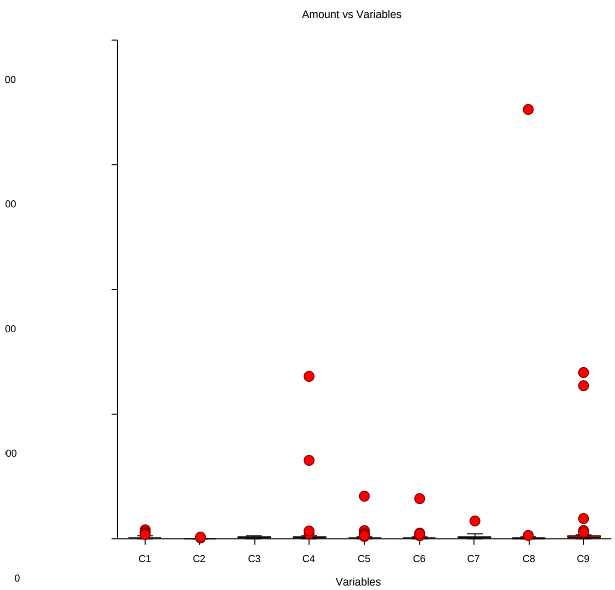
Tests of the Normality of Residuals Assumption

Normality Attributes	Test Value	Prob Level	Reject Normality? (α=0.20)
Skewness	19.3315	0.00000	Yes
Kurtosis	12.2369	0.00000	Yes
Skewness and Kurtosis (Omnibus)	523.4485	0.00000	Yes

Tests of the Equality of Group Variances Assumption

Test Name	Test Value	Prob Level	Reject Equal Variances? (α=0.20)
Brown-Forsythe (Data - Medians)	0.9030	0.51437	No
Levene (Data - Means)	3.6834	0.00042	Yes
Conover (Ranks of Deviations)	230.2064	0.00000	Yes
Bartlett (Likelihood Ratio)	752.1719	0.00000	Yes

Box Plot Section



One-Way Analysis of Variance Report

Dataset	Untitled
Response	C1,C2,C3,C4,C5,C6,C7,C8,C9

Expected Mean Squares Table

Model Term	DF	Term Fixed?	Denominator Term	Expected Mean Square
A (...)	8	Yes	σ^2	$\sigma^2 + sA$
Error	279	No		σ^2

Note: Expected Mean Squares are for the balanced cell-frequency case.

Analysis of Variance Table and F-Test

Model Term	DF	Sum of Squares	Mean Square	F-Ratio	Prob Level	Reject Equal Means? (α=0.05)	Power (α=0.05)
Between	8	4.319788E+07	5399735	0.9049	0.51277	No	0.41984
Within (Error)	279	1.66491E+09	5967420				
Adjusted Total	287	1.708108E+09					
Total	288						

Welch's Test of Means Allowing for Unequal Variances

Model Term	Numerator DF	Denominator DF	F-Ratio	Prob Level	Reject Equal Means? (α=0.05)
Between Groups	8	112.58	5.0419	0.00002	Yes

Kruskal-Wallis One-Way ANOVA on Ranks

Hypotheses
H0: All medians are equal.
H1: At least two medians are different.

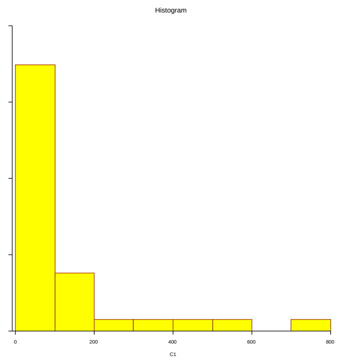
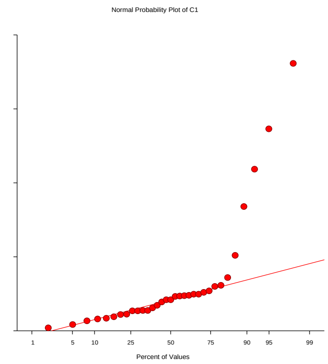
Test Results

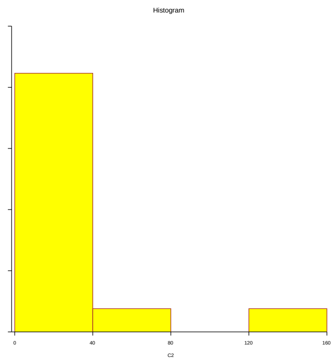
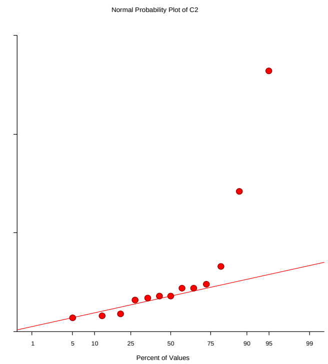
Method	DF	Chi-Squared (H)	Prob Level	Reject H0? (α=0.05)
Not Corrected for Ties	8	30.2302	0.00019	Yes
Corrected for Ties	8	30.2327	0.00019	Yes

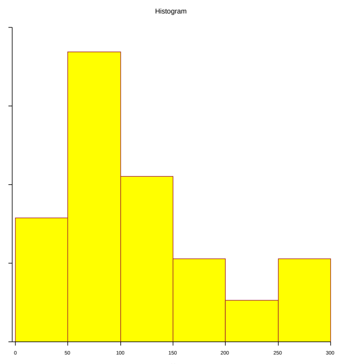
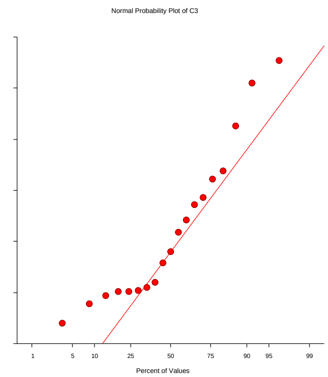
Number Sets of Ties	65
Multiplicity Factor	2004

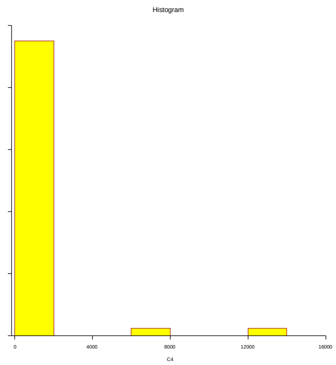
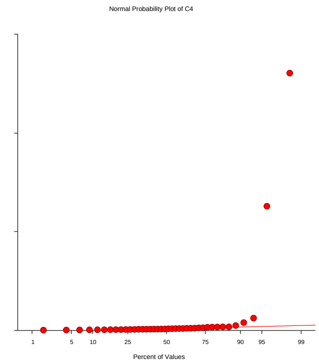
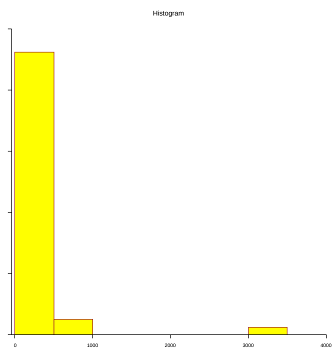
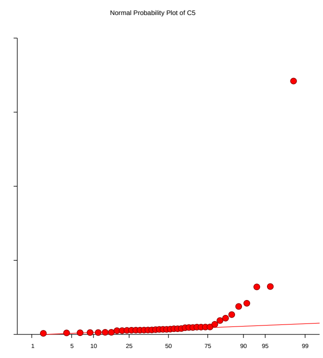
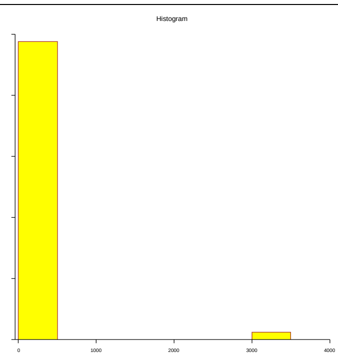
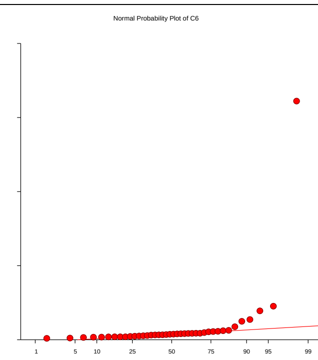
The following tables describe the Total variables and the Unit errors per variable attributes of the models by year.

<table>
<tr><th colspan="3">Variables C1 though C9 are 2009 through 2017 total variables</th></tr>
<tr><td>2009

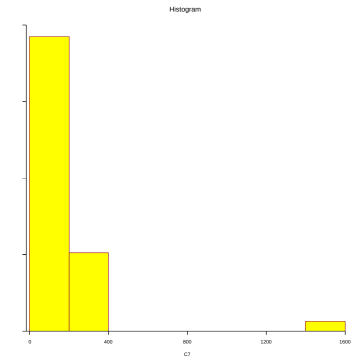
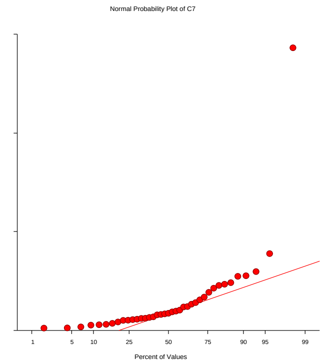
33 models</td><td>
</td><td>
</td></tr>
<tr><td>2010

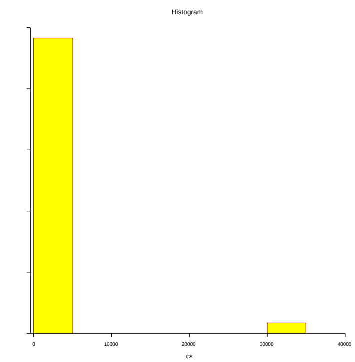
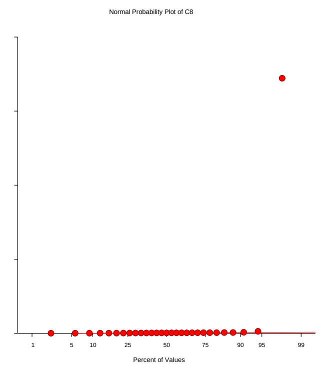
13 models</td><td>
</td><td>
</td></tr>
<tr><td>2011

19 models</td><td>
</td><td>
</td></tr>
</table>

Variables C1 though C9 are 2009 through 2017 total variables		
2012 40 models		
2013 40 models		
2014 40 models		

<table>
<tr><th colspan="3">Variables C1 though C9 are 2009 through 2017 total variables</th></tr>
<tr><td>2015

39 models</td><td>
</td><td>
</td></tr>
<tr><td>2016

29 models</td><td>
</td><td>
</td></tr>
<tr><td>2017

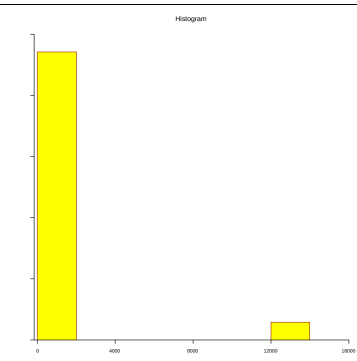
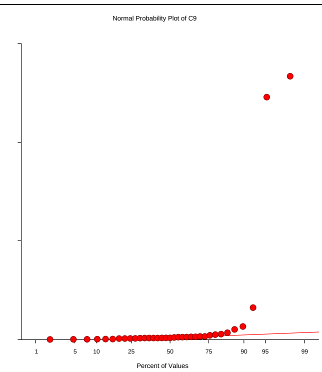
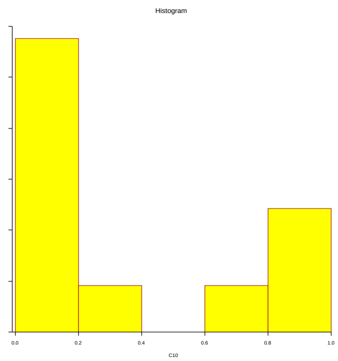
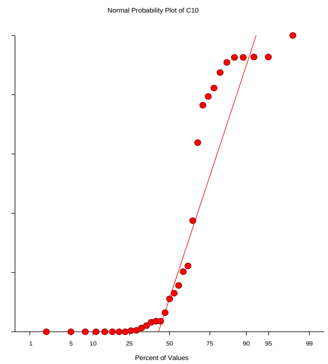
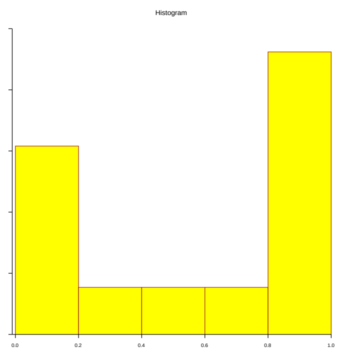
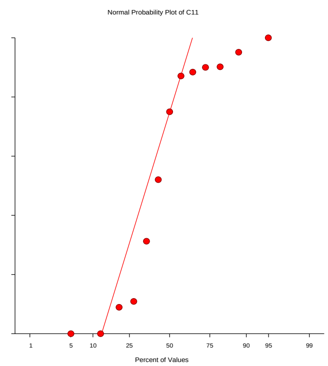
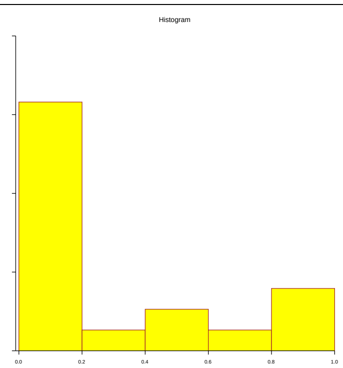
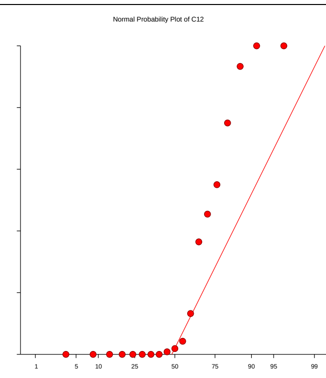
35 models</td><td>
</td><td>
</td></tr>
</table>

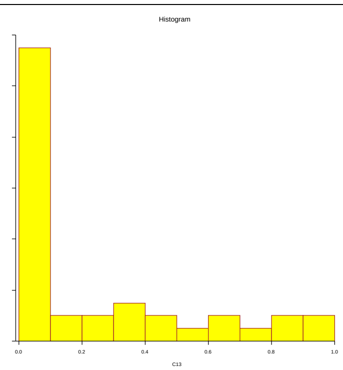
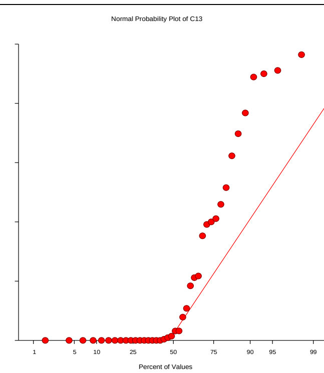
Table 9 Variables C1 though C9 are 2009 through 2017 total variables

<table>
<tr><th colspan="3">Variables C10 though C18 are 2009 through 2017 of unit errors per variable</th></tr>
<tr><td>2009

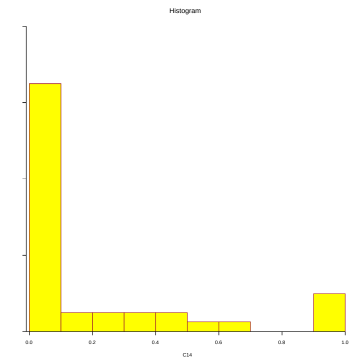
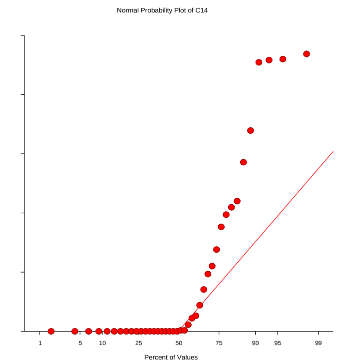
33 models</td><td>
</td><td>
</td></tr>
<tr><td>2010

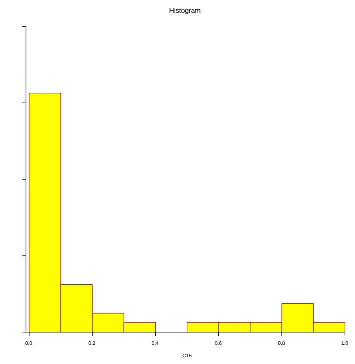
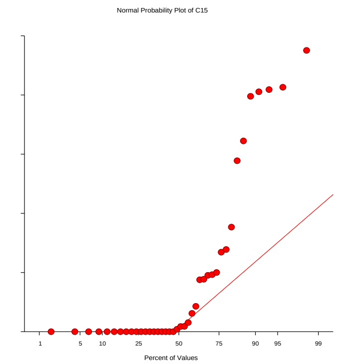
13 models</td><td>
</td><td>
</td></tr>
<tr><td>2011

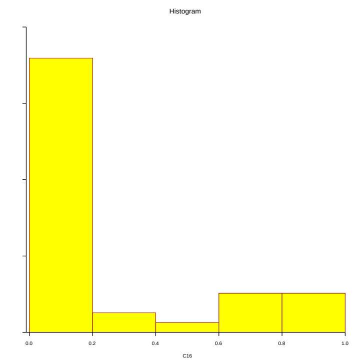
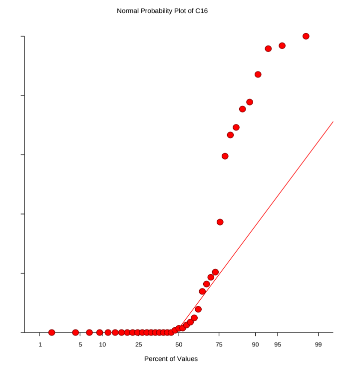
19 models</td><td>
</td><td>
</td></tr>
<tr><td>2012

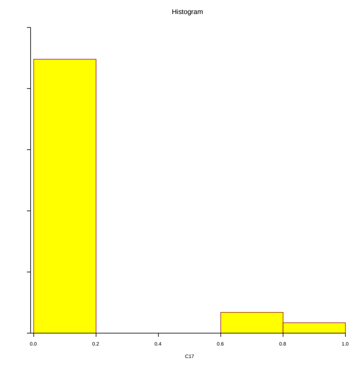
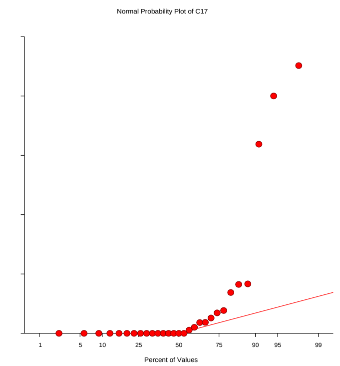
40 models</td><td>
</td><td>
</td></tr>
</table>

<table>
<tr><th colspan="3">Variables C10 though C18 are 2009 through 2017 of unit errors per variable</th></tr>
<tr><td>2013

40 models</td><td>
</td><td>
</td></tr>
<tr><td>2014

40 models</td><td>
</td><td>
</td></tr>
<tr><td>2015

39 models</td><td>
</td><td>
</td></tr>
<tr><td>2016

29 models</td><td>
</td><td>
</td></tr>
</table>

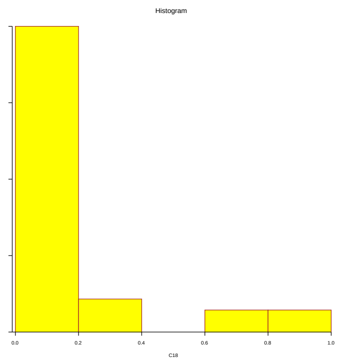
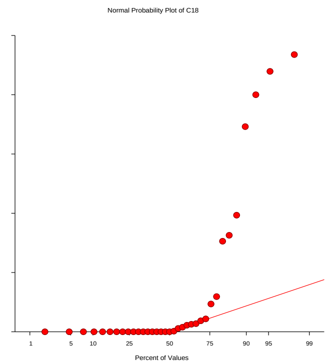
Variables C10 though C18 are 2009 through 2017 of unit errors per variable		
2017 35 models		

Table 10 Variables C10 though C18 are 2009 through 2017 of unit errors per variable