Data-Driven Asset Management

Possible finally matches vision

Richard G. Lamb

Chapter 2
Data, Analytics and Software to be Data-Driven

Version 1/7/2020

ISBN 978-1-7343947-0-2

Contents

Chapter 2
Data, Analytics and Software to be Data Driven

Data-driven asset management is defined as using the firm's operational data to augment the experience and judgement of its operatives, managers, analysts and engineers as they plan, organize, conduct and control the functions, processes and resources of operational availability. The difference between data-driven and traditional asset management is that "possible finally matches vision."

Only by being data-driven is it possible to drill into the top-level factors of operational availability, discover and subject what matters most to data-enabled analyses and reengineer for better outcomes. Only by being data-driven is it possible to confirm that the reengineered outcomes are truly shifting achievable availability upward and moving it toward its peak while reducing the gap between achievable and operational availability. Only by being data-driven is it possible to assure that all is taking place that must take place daily, weekly, monthly and annually to reach and sustain greater and cost-effective operational availability.

At this juncture it is necessary to establish an initial understanding of data, analytics and insight. Accordingly, the chapter will introduce what we are trying to do and the methods and software with which they are done. The remaining chapters will dive much deeper into what is introduced in this chapter.

2.1. Big Picture

Let's draw a big picture of the what data-driven asset management looks like. To begin, the section depicts a data-driven operation. Next, it will introduce the critical-mass knowledge, skills and software to reach the

depicted operation; virtually without cost and up from the grassroots. And finally, the section will establish the basic definitions of data and analytics and, thus, separate the meaningful few from the distracting many.

2.1.1. What It Looks Like

Data-driven asset management was defined in the opening paragraph to the chapter. Figure 2-1 depicts a data-driven operation or process.

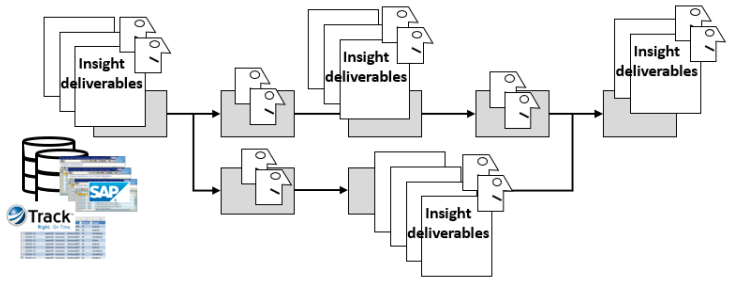

Figure 2-1: What a data-driven operation or process looks like.

What we see is that the processes of a data-driven operation are improved in a particular way. The judgement and experience of the role holders along the process are augmented with insight deliverables.

All activities to beget and manage operational availability flow along the processes of data-driven asset management. At some places along the processes, the best outcomes can only be realized when experience and judgement are augmented with insight deliverables. At each such place, the value of one or more insight deliverable is recognized, built and worked to realize the best of outcomes.

This implies a criterion for any developed insight. Each insight is justified if it makes a difference for the firm's financials and return on investment.

Also depicted in the figure is that the insight deliverables are built upon the data captured in the firm's operating systems. This is because the systems across an enterprise collectively capture every data point generated in the conduct of the collective operational processes.

Furthermore, modern systems are designed to make it easy to retrieve their data as standard reports.

Also depicted is that some data may be captured in Excel tables. These tables can be positioned for ready access.

2.2.2. Critical Mass and Grass Root

Say “data-driven” to management and they cringe. They envision a complex, high-tech, deep-capital initiative and every horror that goes with it. However, this is a gross misperception. It has been propagated by purveyors and media as they speak of grand and glorious initiatives.

The reality is opposite to the perception. Almost every imaginable insight deliverable to an operation can be built and managed at the grass-roots. This is because the “critical-mass” to becoming data-driven is not high-tech or new-tech. It is the exercise of modern-day knowledge, skills and software. The book is written to explain the critical-mass of data-driven asset management rather than stories of the grand and glorious.

Critical-mass is defined as the threshold knowledge, skills and software that must be in place for an operation to be fully, effectively and efficiently data-driven.

Critical mass has two qualifying characteristics. First, should they happen, the threshold knowledge and skills will travel to up-teched and up-scaled strategies. Second, up-teching from critical-mass will not practicably increase the power of the insight that is extracted from the operation’s data.

What is critical mass for data-driven operations is such that the most subordinate processes can be reengineered as a grass root initiative. This is because critical-mass rests upon a triad of software which is already normal to our work and organizations or which we have free rights to them. Figure 2-2 shows the triad to almost any given insight deliverable.

The data to all insight deliverables are readily accessible from the operating systems and Excel files that have captured them. When located, they are extracted from their sources and joined together in a super table with MS Access.

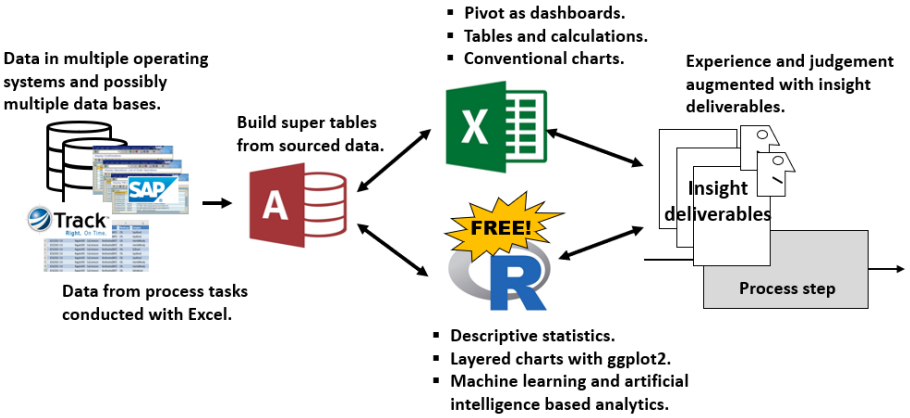

Figure 2-2: The critical mass triad of software to data-driven operations.

Then we will go down one of two paths for each insight deliverable. One is the path to Excel to build dashboards. Alternately, we may go down the path to subject the data to analytics using the free, open-system software known as "R." By whichever path, there is an insight deliverable at the end of the trail.

This is good place to make a point with respect to grass-root and critical mass. Remember the expression, "Systems talking to each other?" There has been a great deal of progress over the decades toward the vision. However, we are still far from the prerequisite degree of systems integration needed for unconstrained data-driven asset management.

However, in Figure 2-2 we can see that there is de facto systems integration. The constraints to data-drivenness no longer exist.

This because systems "talk to each other" through their data bases. Reaching for data from any available source and building super tables constitutes pseudo systems integration for data-driven asset management. In line with the principle of grass-roots, the integration is achieved with easily learned skills rather than big capital.

Of course, there are commercial alternatives for each software of the triad. However, we must be sure that the difference is more than just "prettier." An organization may opt for a commercial alternative for strategic reasons. However, short of some strategic rationale, everything a commercial offering can do can be done by the triad.

The ramifications and distinction for critical-mass-supported insight deliverables are clear if we compare the software triad to the industrial internet-of-things (IIoT) for condition-based maintenance (CBM). The grassroot triad of Figure 2-2 entails no infrastructure beyond what is natural to any organization. In contrast, IIoT-supported CBM, as shown in Figure 2-3, requires infrastructure in addition to a firm's existing infrastructure.

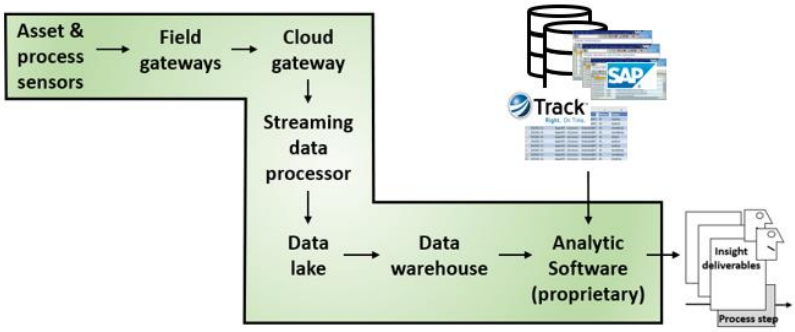

Figure 2-3: Insight deliverables built upon IIoT infrastructure.

New sensors are placed on the process or asset to continually monitor asset and process characteristics. The data of the sensors flow through new gateway infrastructure. Thence, there is additional new infrastructure to deal with the massive streaming, disparate data from the sensors and transforming them to workable structured form. Finally, there are new proprietary analytic software to conduct the analytics of the envisioned insight. Data may also be extracted from the firm's operating systems and pulled into the analytics.

This is a good place to accentuate a point, In the domain of asset management, CBM is the only type of insight that may require a system akin to Figure 2-3. Therefore, it is extremely important that we do not allow CBM to be mistaken by management as the essence, definition and overarching purpose of data-driven asset management.

In turn, it is also important that we do not allow IIoT-supported CBM to be mistaken as on-condition maintenance. CBM is one of four alternatives for an on-condition maintenance task to a failure mode—CBM, product quality, primary effects and inspection. The choices are

made on worth with respect to safety, environment, operational capability and collateral damage.

Taken a step farther, IIoT-supported CBM is a choice to automate some or all aspects of a CBM solution. Consequently, the issue is the relative worth of IIoT-supported CBM in contrast with conducting CBM tasks with the many less grand and glorious offerings to achieve the same end.

Moubray's experience is that CBM is feasible for 20 percent of failure modes and worth doing for less than half of them. The four categories of on-condition maintenance are suitable for 25 to 35 percent of failures modes.[1]

Because IIoT-supported CBM entails the extreme shown in Figure 2-3, it will rarely be a large-scale choice. It may be a worthwhile for a percent or so of cases. Consequently, it is important to not be distracted from the fact that almost 100 percent of insight deliverables to asset management can be done with the triad. For them, the worth is huge, and the cost is almost nothing other than the enterprise's commitment to learn new skills with standing software.

2.2.3. Essential Definitions

There is tremendous hyperbole around data and analytics. Ironically, the hyperbole may be a cause for managements' instinctive aversion to the discussion of data-driven operations. It has also caused tremendous confusion as an obstacle to becoming data-driven. To get beyond the aversion and confusion it is now necessary to narrow the hyperbole to the meaningful few definitions. With respect to them, the terminology of the hyperbole are expressions of the same thing but with different and flashy wording.

The essential definitions can be narrowed to four. They are data and big data, machine learning, artificial intelligence and algorithms. Many terms have come and gone over the last several years, but the four will remain as the language of data-driven asset management.

[1] Moubray, John. Reliability-Centered Maintenance. Second edition. Industrial Press. 1997. Page 155

Data and Big Data. Data and big-data are distinctively different with respect to necessity, technology and organizational abilities. It is an important distinction. This is because big data entails high-tech systems and infrastructure, specialized skills and capital. Data does not.

We tend to think of "big data" in a colloquial sense from working with Excel. In the context of Excel, thousands or hundreds of thousands of rows are "BIG." It is impossible to work with that much data in an Excel worksheet. Things are laborious once we get beyond a few hundred rows.

However, lots of data does not make it big data. Big data is the case in which data are prohibitively massive or unstructured. A professor of data science proposed a simple litmus test of data versus big data. It is big data if the data or the analytics of the data cannot be work on our notebook computers.

Unstructured data include disparate types of data such as streaming sensor data, e-mail, document, video, photo, audio and webpage. This is compared to the structured alpha and numeric data that is predominate to operating systems. The purpose of analytics of unstructured data is to transform them to be structured data with which other analytics can be conducted.

The types of data analytics conducted in either arena are the same. The type does not decide whether it is data or big data. However, the important notable reality for data-driven asset management is that there are very few imaginable needs for big data.

Let's look at a situation that is not big data, but the definition of unstructured data may cause us to assume it is. Other than the massive, streaming data of CBM systems, the only unstructured data in most operating systems are the free-text variables of work order descriptions and notes. Text mining analytics may be used to classify the work orders by the failure mode that triggered them. Remembering the professor's litmus test, although free-text is unstructured, it is possible to conduct text mining on our notebook computers.

The IIoT-supported CBM of Figure 2-3 is an example of big data. Streaming data is massive because the data points are almost infinite in

number. Additionally, for some types of monitoring, the data is not structured data and must be transformed by analytics before it can be subjected to the ultimately planned analytics.

Therefore, when the expression "big data" is casually tossed about, we must ask an inspectorial question. Is it hyperbole or big data? If it is truly big data, then we cannot take the grassroot strategy as it is otherwise possible with the triad of software.

And once again, a very important point must be repeated. Almost never will the insight deliverables to asset management entail big data. This is a fundamental reason that data-driven asset management can be grown from the grass-roots.

Machine learning, artificial intelligence and algorithms. We need to clarify the interrelated terminology of machine learning, artificial intelligence and algorithms. We will use the two-variable regression analysis shown Figure 2-4 as a frame of reference.

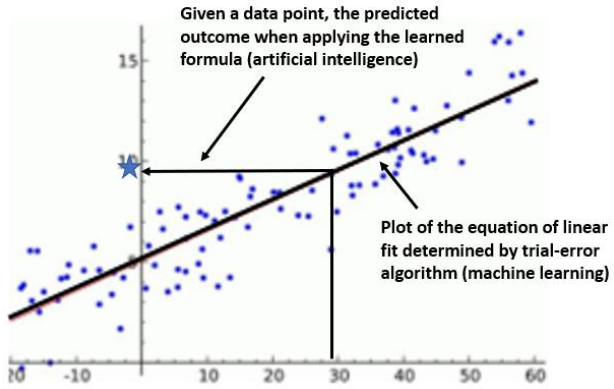

Figure 2-4: A two-variable regression demonstrates machine learning, artificial intelligence and algorithms.

We have all at some time touched regression modeling. However, the concept is the same regardless of the type of model and the number of predictor variables placed in the model.

Machine learning (ML) takes place when we feed the predictor and outcome variables to the regression. The gut algorithm conducts a trial-and-error calculation until "learning" the best fit and returns a formula of an intercept and slope coefficient. The predictive and outcome

variables are the axes and the points are their cross-plots. The line fit to the plotted points is the learned outcome of the algorithm.

Most often our interest ends with the returned coefficients for each variable (one in this case) and associated inferences for how strongly, if at all, the predictor variable is related to the outcome variable. Confidence interval to the coefficient will also get our attention. In contrast, artificial intelligence (AI) feeds new cases to the fitted model to predict outcomes upon the "learned" formula.

AI does not distinguish the model. All types of models entail machine learning, and most can be deployed as AI.

When the model is to be deployed as AI, the learning process will entail an additional stage of analytics. A portion of the original data set is held out from the machine learning stage to be a test set. The remaining portion is fed to the model for learning. The test set is subsequently fed to the learned model to evaluate how accurately the "trained" model estimates or classifies the actual outcome of each case in the test set.

If accuracy is acceptable to the intended use, the model is deployed to serve its purpose—augment human experience and judgement. Results better than 85 percent are typically considered acceptable. This is why we must always think of AI in terms of "augmenting" rather than "supplanting" experience and judgement. Hyperbole may lead us to unrealistically expect "supplant."

2.2. What and Why of "R" and Access

We are all experienced users of Excel. In contrast, the awareness of "R" is rare and experience with Access is unusual. Consequently, now is the time in the explanation of data-driven asset management to introduce the analytic core, "R," and means to move data from its source to its use, Access.

The section will be an overview of the "R" and Access software. The discussion of "R" will be extended to introduce what the book calls layered charting. Written for self-directed learners, the next two chapters will dive deeper into both software.

2.2.1. "R" for the Analytic Core

Something fascinating has happened. Software have become available that are open systems and freely available to any individual to install on their computer and any organization can make part of its IT system. They are also being pulled into commercial software.

These software are not being offered under some sales strategy such as a "trial period" and a "free" compared to a "professional version." Nor are they weak compared to their strongest commercial competitors.

The analytic software "R" is one of such offerings. A testimonial to its strength and unrestricted accessibility is that Tableau and Power BI have seamlessly incorporated "R" into their offerings rather than develop a comparable proprietary capability.

"R" can be downloaded and installed from the website https://www.r-project.org/. There are YouTube videos explaining the simple download process which takes less than 15 minutes.

Other than full-powered, open and free, there are additional reasons that make "R" critical-mass to data-driven asset management. The pinnacle aspect is that through "R" the asset management organization gets it capability for descriptive statistics, layered charting, data cleansing, and machine learning and AI.

"R" is actually a collection of thousands of "packages" for working with almost every imaginable analytic. Every analytic is conducted with a "function" and its associated "arguments." We identify the packages and functions we need to conduct an envisioned insight deliverable. Thence, we do not write code—we type or paste the function code in our R-session as a go-by and adjust it to purpose.

Below is an example of a package, function and arguments. The function ml() for linear repression is available from the "stat" package. The arguments are designated within the parenthesis of the function. The function and its arguments are. . .

```
lm(formula, data, subset, weights, na.action, method
   = "qr", model = TRUE, x = FALSE, y = FALSE,
   qr = TRUE, singular.ok = TRUE, contrasts = NULL,
   offset, ...)
```

Notice that a function is set up by the choices we make for its arguments. Explanations and examples of the options are readily available from the internet. However, we rarely touch most of the arguments because the defaults are typically the desired option. Accordingly, the shown code may reduce to `lm(formula, data)`.

The packages are created and maintained by individuals and organizations around the world in accordance with standards of creation and care. Each package is accompanied with a full explanation of its functions and arguments. Additionally, the explanation includes examples and data with which we can see them in action and experiment.

An extremely important characteristic of "R" is that online support is highly evolved, vast and free. However, we are not limited to online sources. Literature explaining the principals and methods of statistic and analytics with "R" is plentiful. As they explain the principles of statistics, they demonstrate them with "R." As they do, the texts additionally explain each line of code. Consequently, the texts serve concurrently as texts on analytics and the "R" software.

The bibliography to the chapter includes the best available text for every type of analytic. "Best" is defined as a practical working depth explanation; able to be a go-by. This is compared to deep discussions of underlying theory and mathematics.

This book will often use examples from the bibliography literature rather than examples from asset management operations. The generalized examples will have an obvious go-by fit to the aspects of asset management being discussed. Because generalized examples can be go-by's, it is much more important that the readers have at their avail a full-depth explanation of the analytic rather than a domain-specific one.

The rationale of bibliography-sourced examples is demonstrated by the seeming simplicity of the previous two-variable regression analysis. Setting up and interpreting the model is only a tip of the iceberg. Below the surface there are many matters of selecting variables and validating the model. Covering the full depth of every analytic would make the book a 2,500 or so page venture far beyond anyone's willingness to write or read.

Chapter 3 will present and explain how to work with "R." The explanation of this chapter will be introductory.[2]

Figure 2-5 show the primary three windows of "R." Like all software they can be moved and sized.

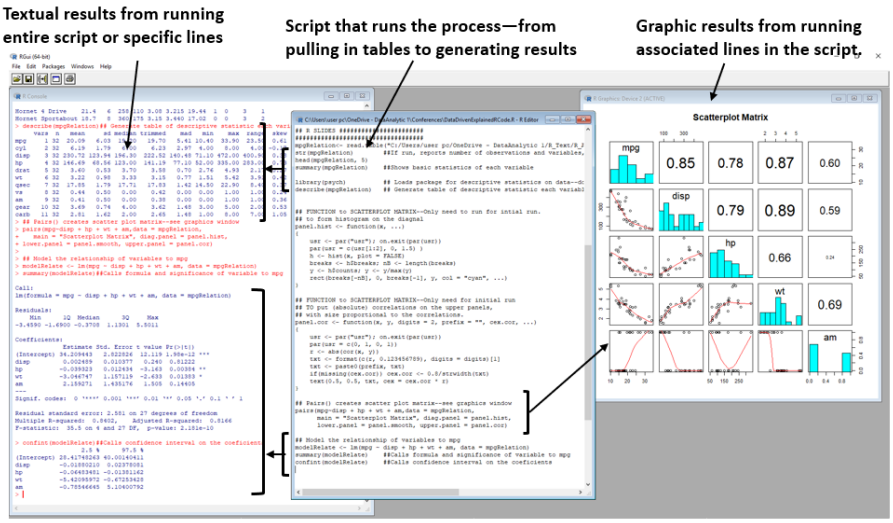

Figure 2-5: Console, script and graphic windows of R.

At the left side is the console window. In it, we can place all commands and get a continuation or output upon pressing "enter" at the end of each line of code.

The center section is the script window. We are not required to use it rather than the console. We do because the window allows us to work with code in a more friendly, flexible manner. Another reason is that what is coded can be saved as a script file.

The difference between the script and console windows is that commands are typed in the console for each occasion, whereas, they can be edited and run repeatedly from the script window. Otherwise, the

[2] de Vries, Andries and Meys, Joris. R for Dummies. John Wiley & Son, Inc. 2015.

Grolemund, Garret and Wickham, Hadley. R for Data Science, O'Reilly Media, Inc. 2017.

Matloff, Norman. Art of Programing R. No Starch Press. 2011.

difference is that the output of running the script appears in the console and graphic windows but never in the script window.

If the script or console code contain commands for graphic outputs, they will appear in the graphic window. Obviously, it is the right-most window of the figure.

The code of the script window can be commanded to run in its entirety or by highlighted lines. Just as valuable, the script allows a solution developed by one person to be distributed to others as a script file. Let's note here that if the script file name is extended with .txt, it becomes a text file, able to be read and edited as a Notepad file.

Although using code may appear to be geek-like scary, coded software has a big advantage over the graphical user interface (GUI) software we have grown accustomed to. Dispersing a GUI-based solution requires a lengthy instruction document and all the difficulties that entails. Scripts can be dispersed as a file of code with explanations placed in the code. Just as important, the recipient does not need to follow a documented instruction as one does for GUI. Instead, the user only needs to load and run the script.

The first line of shown script code is a function that imports the data to the planned analytics. Beneath that, other functions explore and inspect the data in text and graphic form. At the bottom are the functions to the analytics we seek. The script's content will be fully presented and explained Chapter 3.

2.2.2. Layered Charting

Now is a good time to introduce an important breakthrough to insight—layered charting. Layered charting is a big leap in the ability to extract visual information from data. The "R" package to create layered charts is ggplot2.[3]

Most data are still visualized with types of charts invented as far back as the 1600s and no more recently than the 1800s. Now layered

[3] Wickham, Hadley. ggplot2: Elegant Graphics for Data Analysis. Second edition. Springer.

charting allows the visualization of data as information in almost endless ways.

The book defines layered charting as presenting information in layers. The difference can be seen in Figure 2-6. Traditional and layered charting are shown side-by-side.

Characteristic	Conventional	Layered
Types	limited	Unlimited
Number of variables	Two	Unlimited
Number of perspectives	One	Unlimited
Legends	Upon the variable	Can be variables
Data points	Lose distinction if many	Methods to create distinction

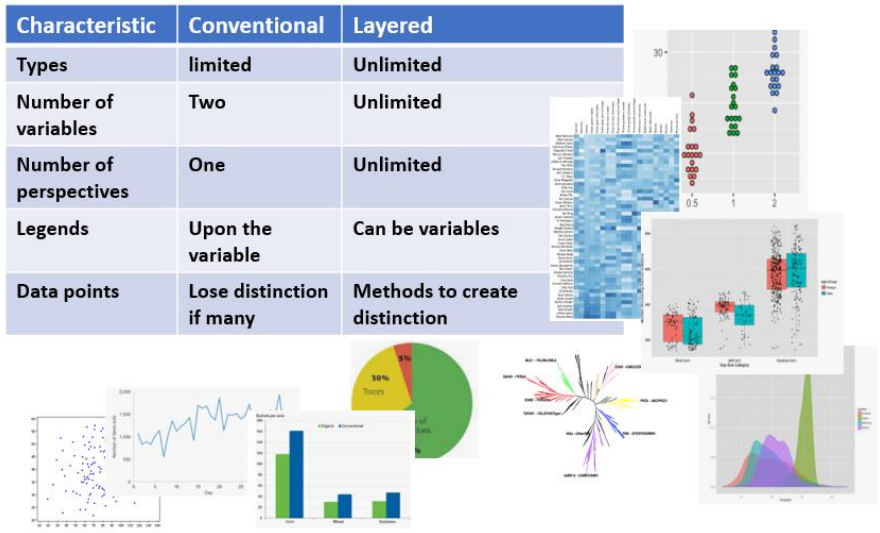

Figure 2-6: Contrasts between traditional and layered charting.

The matrix of the figure summarizes the differences. For one, traditional charting is limited to the named types such as cross plot, bar and column, pie, line, spider, etc. The layered charts have no type or name because they are named by the insight they give.

Traditional charts are limited to the two variables of the axes. In contrast, the number of variables that can be pulled into a layered chart is only limited by practicality.

Traditional legends are limited to the categories of the charted variables. Layered charting can use legends as variables.

Another important differentiation is the capacity for a large number of data points. Visual granularity is lost when there are too many data point. Layered charting offers many ways to regain granularity. Some are shown in the figure.

Figure 2-7 is an example of layered charting built upon the variables by which cars are evaluated and compared. It demonstrates possibilities for presenting the KPIs of asset management.

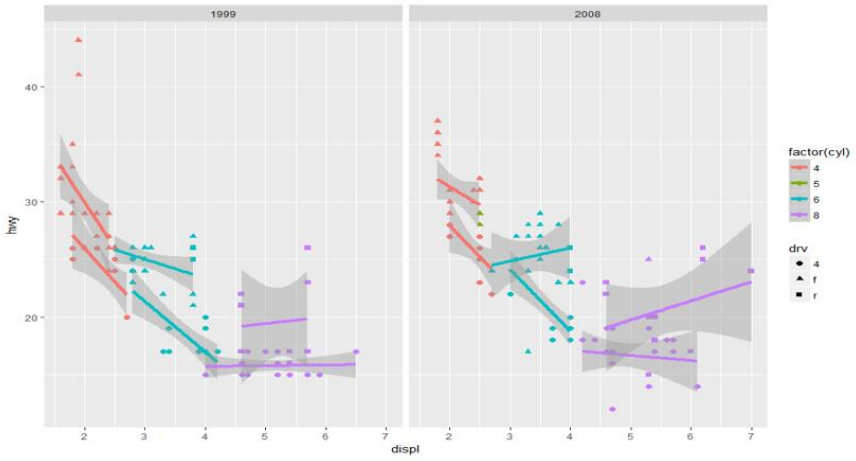

Figure 2-7: Layered charting to present measures of performance.

In the figure, we see the interplay of five variables: highway mileage, displacement, cylinders, drive and years. All are related to the axes of displacement and mileage. Accordingly, we see a cross plot relationship of the two variables with a linear fit and confidence intervals to the fit. Traditional charting would show a negative relationship—mileage falls as displacement increases.

However, the shape of the points would suggest that there is more to the picture. A traditional two-variable cross plot and linear fit may be misinformation.

The cylinders and drive are layered into the chart using legends as the method. A new picture emerges. The relationships vary with the added variables. In some clusters the relationship is positive rather than negative. When year is layered into the chart, it is revealed that the relationships have changed with time. One relationship has even changed direction.

2.2.3. MS Access for Super Tables

There can be no dashboards and analytics without a table inclusive of the variables they are to be built with. This book will call them super tables in contrast to the sub tables that are combined in the super table.

Super tables can be built in "R" with the structured query language (SQL) capability and with additional coding in the rare occasions when it is necessary to go beyond the ability of SQL. The variables from their sources are imported into Excel Pivot and "R."

However, this requires substantial skills with SQL and "R" coding. That is why MS Access is one of the triad software.[4] The skill requirements shrink to minor and largely entail skills that have become normal to most of us. Therefore, the more practical strategy is to learn how to build tables in Access and then import them to "R" with the "read" function.

This section will introduce the concept and process of building super tables. The Chapter 4 will go deeply into the how-to of super tables.

Figure 2-8 is a pictorial summary of what is to be done. Any insight deliverable requires all needed variables in a single table. The table must be formatted with variables as columns and cases as rows. There are no row titles or space and sum lines. All tables in the figure meet the standards.

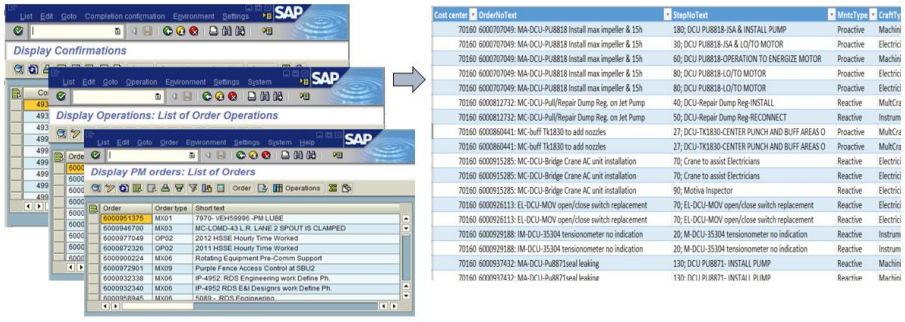

Figure 2-8: Three sub tables combined in a single super table.

[4] Alexander, Michael and Kusleika, Richard. Access 2016 Bible. John Wiley & Son, Inc. 2016. Parts 3 and 4.

The figure shows the typical hurdle to building insight deliverables. The needed variables exist in three sub tables. They must be brought together in a single super table. Otherwise, asset management is divided and conquered by the firm's operating systems.

Another perspective is that the needed super table does not, cannot and never will exist in any operating system. Furthermore, for those who say, "I would do it in Excel," there are four points to make.

First, building the envisioned table in Excel is too laborious and limitations to be practical. Second, it was said earlier that very quickly a block of data becomes "big" relative to working with data in Excel. Third, Excel is limited to 1.1 million rows of data. Finally, on a personal level, we need to stay modern if we want to stay relevant.

There is a process for building super tables. It is shown in Figure 2-9.

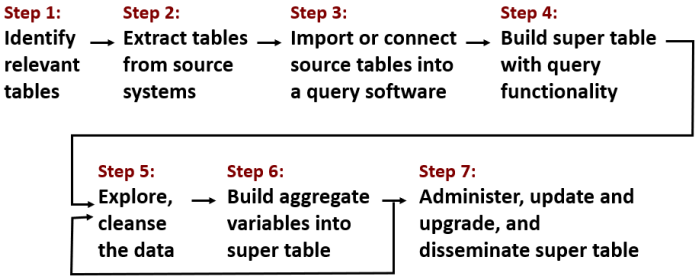

Figure 2-9: The process to build super tables.

The first step is to locate where the variables of interest reside across the enterprise. Once found, the second step is to identify the standard reports by which to extract the data from their resident operating systems. The third step is to bring them into a query software such a MS Access.

Figure 2-10 shows the action to be taken. The tables of Figure 2-8 are standard reports that, once recognized, are imported into Access. The figure also show that two other non-system tables have been pulled in the query. They were built to make it possible to categorize and clarify the data in ways that were not, and never will be, configured into the home systems.

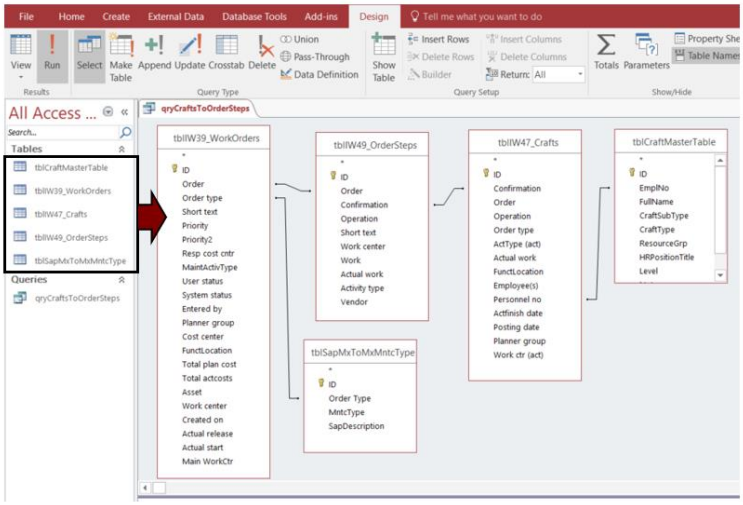

Figure 2-10: Tables imported to the query software and joined as a single table.

Notice the lines between the tables that were pulled into the work area. The line symbolizes that the tables are joined in a massive raw table. Each pair of tables is joined by a unique identifying variable they have in common.

The super table is built in step three. Variables are selected to be in the table as shown in Figure 2-11. By click and drag all desired variables from the joined tables are pulled into the evolving super table That is the purpose of the design grid at the top of the figure.

A lot goes on in the design grid. However, we would find that almost all of what is done draws upon the skills most of us have accumulated throughout our working lives. When the run icon (not shown) is clicked, the super table shown in the bottom part of the figure is generated.

Until confirmed, it is never assumed that the data pulled into the super table is accurate and complete. The next step is to explore the data for issues needing a treatment strategy.

Most times there are simple solutions such as translation tables to be introduced in Chapter 4. In other cases, the table may be pulled into "R" for cleansing with machine learning and artificial intelligence. For

some insight analytics, there is a choice to omit bad data after evaluating the ramifications to the subject insight.

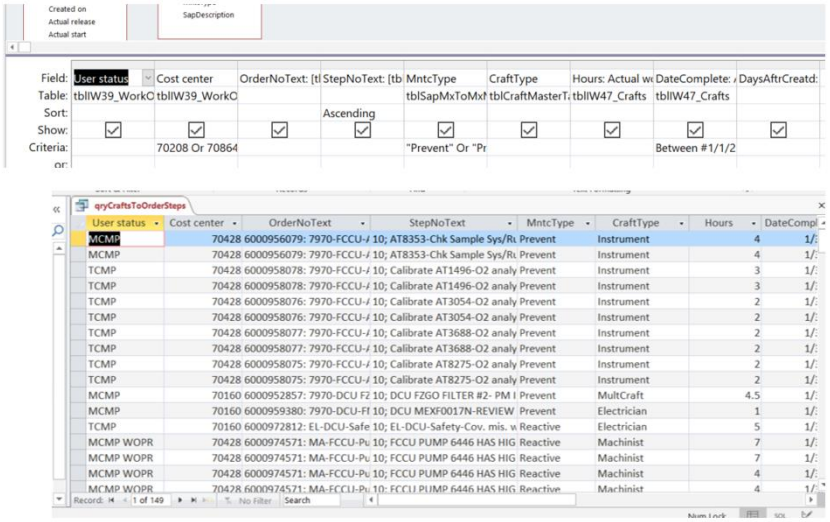

Figure 2-11: Variables pulled into the super table molded for insight.

The sixth step is to build aggregation variables in the super table. The idea is to create new variables in the table. They are totals, counts, averages, standard deviations, , min-max and first-last for groups created upon a set of predictor variables.

All sorts of insight variables are possible when the super table is extended with aggregations. An example is to generate the computed variables for workload-based budgeting and control on actual versus budget. Dual-dimensional budget and variance will be a subject of a later chapter.

Steps two through six are done with standard query language (SQL). The only exception is that some types of cleansing require data analytics. This suggests that one hurdle to data-driven asset management is to grow SQL skills across the organization.

How the hurdle is jumped is the reason for MS Access in the triad of software. This is because SQL runs in the background as we work at the foreground with the skills we all have as modern workers.

Furthermore, besides already part of the Microsoft Office software, it is arguably the easiest of all query software to learn and work with.

Another advantage is that Access stays close to how the sausage is made rather than be hidden from us inside a black box. Accordingly, what is done in the foreground somewhat mirrors the clauses of SQL.

The final step recognizes that any one super table is likely to be built to serve multiple insight deliverables and ad hoc analyses. Therefore, the final step is to form one or more processes to manage each super table through its build and refine, update and disseminate stages. The process may be owned by the primary beneficiary or by someone with the role of building and administering the table on behalf of all players across and beyond the asset management organization.

This brings another point to the surface. The SQL code, automatically formed in the background as we work in foreground, is available as a view option. Consequently, the super table can be distributed as a text file just as for an "R" script. The recipient can paste the text in the SQL view and run it. In turn, the recipient can modify the super table in the design view.

There is a partnership between "R" and Access in the triad. At times we may want to formulate variables or reshape the table in ways that are beyond the ability of SQL. When more is needed, the super table can be built in Access, pulled into "R" and powered up. As previously mentioned, we may also want to subject the table to cleansing analytics that are beyond the ability of SQL.

Some analytics are built with data that must be shaped specifically to a model's algorithm. When the case, the bibliography literature explaining the model will, of course, introduce and explain the "R" functions to reshape the data.

2.3. Insight Deliverables

At the beginning of the chapter insight deliverables were depicted as woven into the processes of asset management. The principle is that the outcomes of the processes would be better than if without the insight.

There are four types of insight deliverables. As named by this book, they are system reported, recountive, know-thy-data and modeled.

Each will be introduced in this section. Subsequent chapters will further expand upon them in the context of explaining how to bring specific sub processes of asset management to be data-driven.

An observation. It is easy to envision data-driven as exotic and grand. However, the majority of insight deliverables are recountive and know-thy-data. This is an extremely important point because the skills to build and work them are of the type that are easily and quickly absorbed and dispersed. Consequently, once we know what we want to achieve in the effectiveness and efficiency of a process, the recountive and know-thy-data deliverables are the easiest of the challenges for getting there.

2.3.1. System Reported and Recountive Insight

System reports are the insights that our systems have been configured to give us. The number of asset management organizations for which system reports are the extent of insight is shrinking.

The number of organizations that function with system reports and recountive insight is growing. This is evidenced by the growing use of Excel Pivot in contrast to spreadsheets in which the data and report are fused. Chapter 5 will introduce the problem and solution to fused data.

In contrast to modeled insight, recountive insight is built directly upon the data of operational systems rather than processed through analytics. Consequently, recountive insight limits the organization to asking and answering questions of who, what, when, where, how much and indicators. In other word they recount history.

As shown in Figure 2-12, recountive insight is packaged and delivered with Excel Pivot or somewhat prettier commercial alternatives. If the data to the Pivot are super tables, the insights are beyond what can be revealed from individual system reports. This is because we are able to slice-dice-drill for insights that not visible in system reports, until their data are joined in a super table.

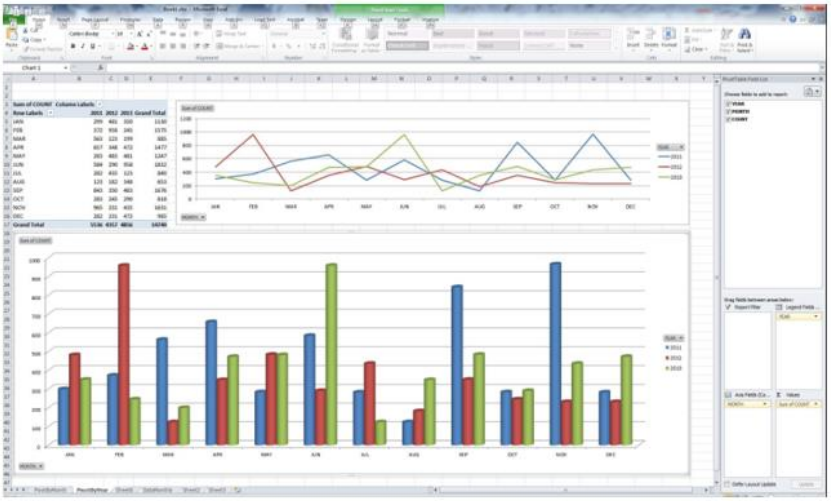

Figure 2-12: Recountive insight deliverables with Pivot of the triad.

We can power-up recountive insight by including layered charting. Excel Pivot and commercial dashboards are largely limited to traditional charting. The combination of conventional and new-age visualization is a good example of the critical-mass analytic software, "R," making it possible to build data-driven processes up from the grass roots.

2.3.2. Know-Thy-Data Insight

One-time and periodic inspections of process data often beget deep insight. One reason is that the persistent compliance to operational processes is readily confirmed through its data. Another reason is that an initial and unfolding inspection of data often reveals hidden realities, findings contrary to common belief and lurking misinformation.

Know-thy-data insight is the case of a table of data explored from descriptive, graphic and statistical perspectives. Figure 2-13 shows some of the possibilities generated with "R." Query-type probing with Access and Pivot are also powerful methods to seeking know-thy-data insight.

The upper right is an example of data variables summarized descriptively when the super table is fed to the summary() function of "R." For each variable we can inspect the min-max, median, mean, quartile, categories, counts and number of missing records.

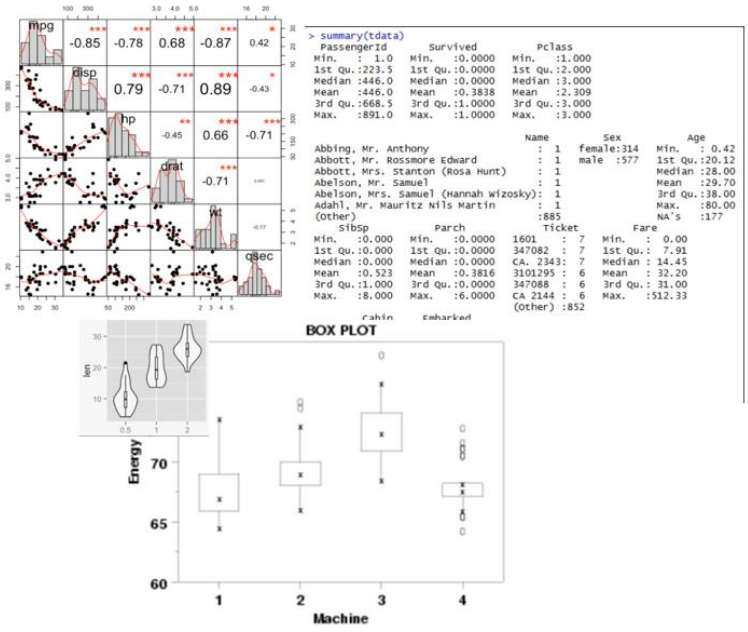

Figure 2-13: Examples of know-thy-data generated with "R."

The lower left is an example of the data presented in graphic form. As previously mentioned, the possibilities for graphic exploration are intriguing because "R" brings non-traditional graphics to the data.

The upper left is a combination of numeric and graphic insight. In it there are eight insights to each variable and the correlations between them.

2.3.3. Modeled Insight

The final category of insight deliverables, as named by this book, is modeled insight. Insight is gained as the data flows through analytic models for machine learning and AI.

Consequently, and in contrast to recountive insight, we can ask and answer five additional types of questions—relationship, difference, time series, duration and apparency. Modeled insight, as duration questioning, was seen in the previous chapter's definitions and explanations of mean time between maintenance (MTBM) and mean time to maintain (MTTM).

Relationship Questions. Which asset and process variables are most strongly related to a performance of interest? Many of us have asked and answered such questions with linear regression.

Figure 2-14 shows that there are three types of regression models. They are linear, logistic and Poisson.[5]

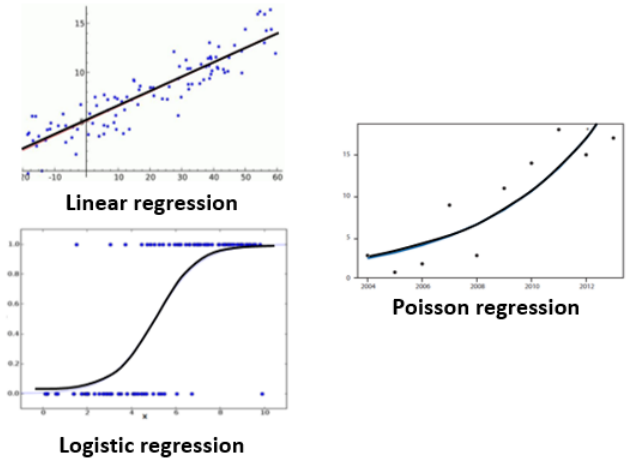

Figure 2-14: Three types of regressions for relationship questions.

A model can entail multiple predictor variables. However, note that single predictor models are shown in the figure so that we can see the underlying fitted shapes of the three outcomes. The shapes hold with two or more predictor variables but are not graphically depictable.

The three types could be subjected to the same sets of predictor variables. The difference is the report-out we seek from the outcome variable.

Linear regression deals with numeric outcomes such as cost, hours, productivity and numeric indicators. The fit is linear, and its outcome can range from positive to negative infinity.

---

[5] Field, Andy and Miles, Jeremy. Discovering Statistics Using R. Sage Publications, Inc. 2012. Chapters 7 and 8.

Finch, Holmes. Multilevel Modeling Using R. CRC Press. 2014.

Zeileis, Achim, Kleiber, Christian and Jackman, Simon. Regression Models for Count Data in R. https://cran.r-project.org/web/packages/pscl/vignettes/countreg.pdf

What if the score is not linear and dually-infinite? What if the fit falls between 0 and 1? In operational processes, many behaviors are binomial—e.g., working or not working. Many others are multinomial—e.g., option 1 or option 2 or option 3.

Logistic regression reports out as the probability of binomial or multinomial outcomes—strength of conviction. For example, in a wrench study, the model can predict the likelihood of a "found-working" outcome. Another is to predict the likelihood an entry is out of compliance with what should have been recorded.

There may be missing or suspicious numerical or classification entries in the super table. Linear and logistic regression, as AI, can be used to impute missing entries and flag suspicious classification.

The Poisson regression models occurrences. They report out as counts or rates. For reliability or process compliance they are respectively probabilities for the number of occurring failures or noncompliances. Rate ties failures and noncompliances to time interval, area and other groupings.

Often there is money in finding outliers. They can make the insights based on them to be misinformation, thus, a payoff upon spotting them. Or they may point us to some phenomenon that, although hidden, reveals intriguing ramifications. The regressions allow us to apply sophisticated methods to find outliers.

Difference Questions. How do slice-dice combinations of asset and process variables comparatively effect a performance of interest?

A whole window of inquiry opens once we discover a fundamental truth. Because two or more numeric outcomes are different does not mean they are different. For example, just because a KPI changes after an improvement program has been implemented does not mean a practicable difference has occurred.

We will use Figure 2-15 to explain the overarching concept of analysis of variance (ANOVA) to engage in this extremely meaningful type of questioning.

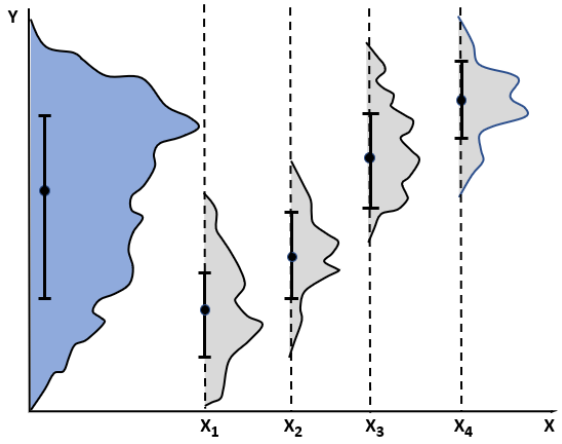

Figure 2-15: The concept of analysis of variance (ANOVA)

Shown at the left, the outcome variable in the model has its statistical mean and confidence interval to the mean. In statistics this is called the base or null model.

What if we grouped the data upon one or more predictor variables? Shown in the figure is a model with four outcome groups. They occurred out of the choice of predictor variables. Each group has its own mean and confidence interval to the mean.

The big question is which pairs are truly different even though their means differ? Do the confidence intervals to the means of the pairs overlap? If so, they are only different due to sampling error.

It would seem that we could answer the question by simply making pairwise determinations. The problem is that what is called "family error" is attached to each pairing. This makes it more likely we would conclude there are differences when there are not.

An example of family error makes the point against simple pairwise analysis. Imagine that the confidence interval established for the analysis is 95 percent. However, the true confidence for each pairing is an unacceptable 74 percent (0.95^6).

What's called the t-test is used if, for example, Figure 2-15 were only a break out of two groups. Comparisons between three or more

groups must be evaluated with ANOVA-type modeling rather than pairwise comparisons.

The determinations are made with post-hoc adjustments and contrasts of variance. With post-hoc adjustments the confidence intervals are made statistically wider (conservative), thus, making it less likely to wrongly accept a pairings as different. Alternately, contrasts of variance evaluate groups by the comparative portion they explain of the total variance of the base (left most) model of Figure 2-15.

Finally, we should note that there are circumstantial variations to ANOVA. They are one-way and multi-way ANOVA, ANCOVA, repeated-measures and mixed ANOVA, and MANOVA. They will be expanded upon in later chapters.[6]

Time Series Questions. What are the components that underlie the summary-level-only history that operating systems are limited to providing? Figure 2-16 shows the primary components of a time series from which the question seeks to gain insight.

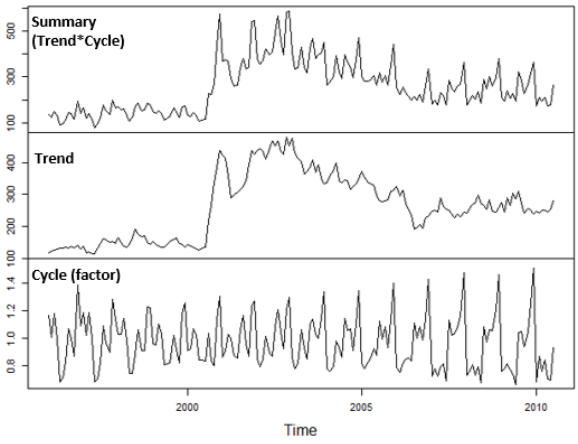

Figure 2-16: Cycle removed from the summary level to reveal the trend of the series.

First it is necessary to extract "cycles" from the summary plot in the top panel. Upon doing so, we can inspect the pattern of cycles. Are

[6] Field, Andy and Miles, Jeremy. Discovering Statistics Using R. Sage Publications, Inc. 2012. Chapters 9 through 14.

they staying steady with time or are they changing? Seen in the bottom panel, the cycle is swinging wider with time beginning with the step up in the trend.

Cycle can be of great interest in a time series. Let's take the occurrence of notifications for needed corrective maintenance. Intuitively, we would expect the notifications to occur randomly. Are we instead observing yearly, monthly and weekly cycles? More importantly, is there a message for asset management in the cycles.

This leaves the "trend." We are interested in its shape and its message for asset management. However, it is necessary to confirm that it is what we think it is. Is it deterministic, random walk or random?

Random is easy to spot. However, a random walk may appear as deterministic. Rather than deterministic, a trend's shape may reflect what is called autocorrelation by which a period is influenced by one or more previous periods.

As an example, let's take a study of actual versus planned craft productivity per scheduled work order. After an initiative to improve job plans and field supervision, is the observed trend in productivity a deterministic one? Alternately, is the trend only reflecting autocorrelation rather than a deterministic input-output relationship?

It is also necessary to question if there is autocorrelation in the variance to the series. Undiscovered, we may otherwise be acting on misinformation. This is because autocorrelation will report out a tighter than true confidence interval for the series.

The principle of autocorrelation has an additional and exciting ramification. The same mathematics can be applied between variables—cross-correlation—to find lead-lag relationships. The trend series of one variable demonstrates a pattern that appears later in another. For example, can we see a pattern in corrective maintenance workload that follows a pattern in plant capacity utilization?

A final point. Prediction is often mistaken as forecasting. However, there is a definitional difference. The distinction is analysis within and beyond the data.

Prediction is what has been explained to this point—within the data. Forecasting uses the findings of prediction to project beyond the data and into the future. The future is being forecast upon what prediction has revealed of the past.

A forecast is built by projecting the trend into the future. The cycle series is also projected into the future and attached to the projected trend. An appropriate confidence interval is placed on the forecast after adjustment for autocorrelation. Of course, we must first question if we can safely assume that the influences of the past will hold into the future.

Finally, we should note here that there are a range of models with which to work with time series. They are Holt-Winter, series regression, ARMA and ARIMA.[7]

Duration Questions. What is the probability an asset or process condition will hold for some time and then what is the probability the condition will end?

The type of questioning was introduced in Chapter 1 to define the reliability and maintainability variables of availability performance. Figure 2-16 shows the two plots—survival and hazard—that were explained in Chapter 1 and a third—cumulative hazard. Because of the earlier attention to survival and hazard, here we will speak to the third panel in the figure.

At some point in time there is the chance of an event (hazard). For example, I am aware that a 72-year-old male has a 2.7 percent probability of passing away in the current year of life. If we wanted to know what to expect over the next five years—cumulative hazard—we would total the hazards at each year of life from 72 through 76.

Incidentally, the expected life (reliability) at 72 is 13 years. Of course, at 72 the meaning of "continuous, trouble free" in the definition of reliability is somewhat nuanced.

A parallel example is "on-condition" maintenance. Having survived to the time of inspection, the inevitability of a functional failure event may be revealed. Once revealed, cumulative hazard will give us a

[7] Cowpertwait, Paul and Metcalfe, Andrew. Introductory Time Series with R, Springer. 2009.

sense of the risk of a functional failure during the time allowed for the orderly corrective maintenance process—mean time to maintain.

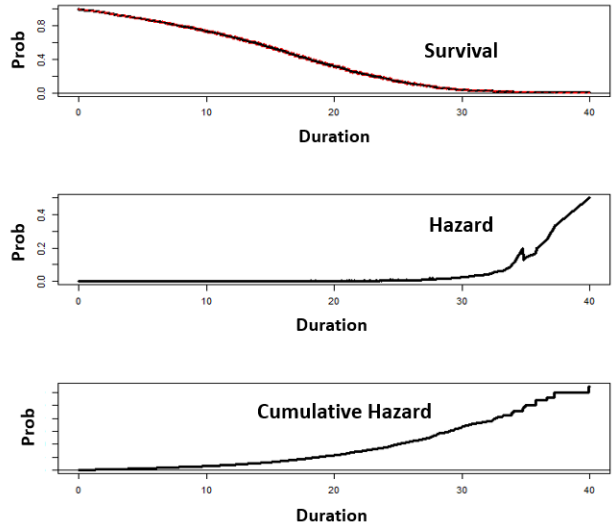

Figure 2-16: The three perspectives of duration analysis.

If the cumulative hazard is too great, we will find ways to shorten the interval of the corrective maintenance process. Duration questioning will play heavily in reengineering the interval. Each stage along the critical path of administrative, logistic and maintenance activities will likely be subjected to duration questioning. The insight will be the shape of the curves—what are they, what can they be and what will we reengineer them to be.

Let's make a side point. The data to reengineer the sub intervals of the maintainability interval are complete and accurate. This is because most computerized maintenance management systems automatically collect status history as a work order passes through its stages from notification to completion.

There is another point to make. The mathematics of hazards make it possible to determine which contextual variables influence the patterns of Figure 2-16. This is exciting because we may find that it is possible to reshape the hazard curve and, in turn, the survival and cumulative hazard curves.

We should note here that there are a range of models with which to work with duration questions. They are Cox regression, Cox proportional hazard, Cox mixed-effects, cumulative incidence, proportional hazard regression, Weibull and Crow-AMSAA.[8]

Apparency Questions. Are there hidden predictor variables to the performance of assets and processes? Figures 2-18 and 2-19 show two of the most popular models to seek out hidden variables—decision tree and K-Means. A distinction between them is to be directed and undirected. A decision tree model is given both predictor and outcome variables. In contrast, a K-Means model is given predictor variables but no outcome variable.

Figure 2-18 is a decision tree model. We seek predictive rules as variables leading to numeric and categorical outcomes.

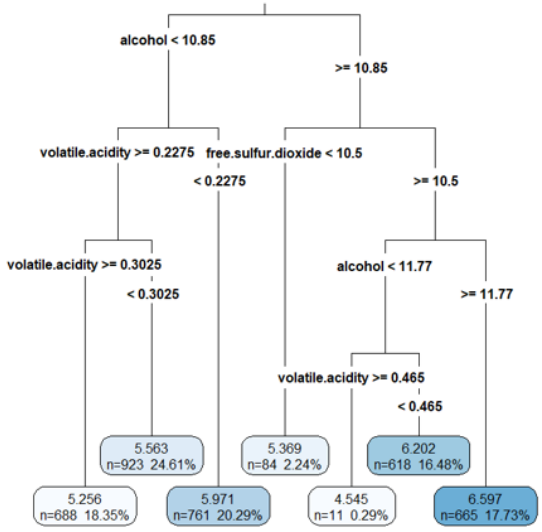

Figure 2-18: Decision tree with numeric outcomes.

At the tree branches, the model determines which one of all provided variables and their splits are most significant to the branching decision. Consequently, the variables are being sliced-diced into variables hidden within them.

[8] Bostrom, Goran. Event History Analytics with R, CRC Press. 2012

A K-Means model also seeks predictor variables by slice-dice. However, there is not an outcome variable against which to decide the slice-dice. Instead, the model reveals clusters of similarity. When the clusters emerge, it falls to the experts in the asset management operation to determine and name what they indicate.

Figure 2-19 shows the case of six hidden variables teased out of two predictor variables. The decision to call for six clusters is determined by an "R" function that evaluates the data to determine the number of significant clusters to seek. However, there is no limit on the number of variables to model other than the interpretative challenge explodes.

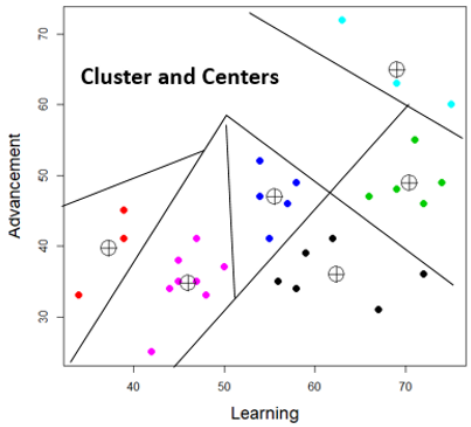

Figure 2-19: Six hidden variables teased out two input variables.

The revealed tree and cluster variables can be joined back in to the super table. The consequence is to tease insightful variables out of the source data that the firm's operating systems cannot. The purpose is to increase the insight power of the models. The variables may also be created to overcome technical problems to a model such as collinearity, too many variables, etc.

There are other models. They variously fall into categories of rule and cluster, and directed and undirected. Later chapters in the book will

identify and expand upon them as they are germane to specific aspects of data-driven asset management.[9]

Bibliography

Alexander, Michael and Kusleika, Richard. Access 2016 Bible. John Wiley & Son, Inc. 2016. Parts 3 and 4.

Bostrom, Goran. Event History Analytics with R, CRC Press. 2012

Cowpertwait, Paul and Metcalfe, Andrew. Introductory Time Series with R, Springer. 2009.

de Vries, Andries and Meys, Joris. R for Dummies. John Wiley & Son, Inc. 2015.

Field, Andy and Miles, Jeremy. Discovering Statistics Using R. Sage Publications, Inc. 2012.

Finch, Holmes. Multilevel Modeling Using R. CRC Press. 2014.

Grolemund, Garret and Wickham, Hadley. R for Data Science, O'Reilly Media, Inc. 2017.

Lantz, Brett. Machine Learning with R. Packt Publishing. 2015.

Matloff, Norman. Art of Programing R. No Starch Press. 2011.

Moubray, John. Reliability-Centered Maintenance. Second edition. Industrial Press. 1997.

Oesko, Suljan. Pivot Tables In-Depth for MS Excel 2016. Suljan Oesko. 2017.

Wickham, Hadley. ggplot2: Elegant Graphics for Data Analysis. Second edition. Springer. 2016.

Zeileis, Achim, Kleiber, Christian and Jackman, Simon. Regression Models for Count Data in R. https://cran.r-project.org/web/packages/pscl/vignettes/countreg.pdf

[9] Lantz, Brett. Machine Learning with R. Packt Publishing. 2015.