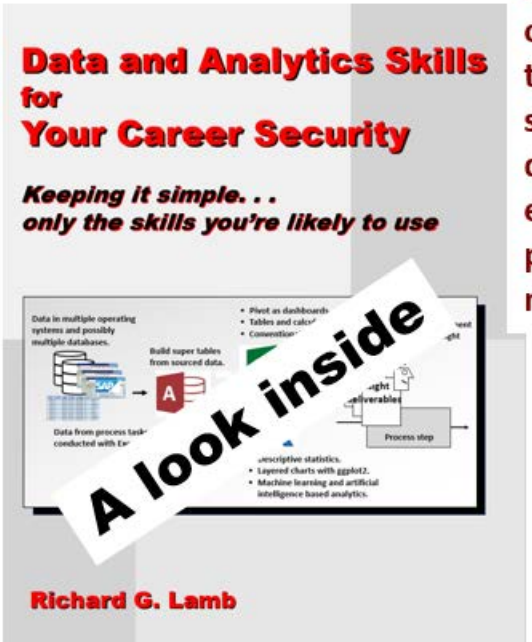

An overview of the Access and R software. The overview of "R" is extended to introduce what the book calls layered charting. Written for self-directed learners, the remaining chapters dive into both software in the context of explaining by demonstration how data-driven planning, organizing and control practices are made possible.

Excerpt:
2.2. What and Why of Access and "R"

Additional "Look Inside" at https://analytics4strategy.com/book-look-inside
Book is available from Amazon
Paperback, 508 pages, 300 figures and outputs, 7.5 x 1.15 x 9.25 inches, 2.37 pounds

Note:

The advance graphics created with the ggplot2 package of the R software depend heavily on color to communicate the insight they are coded to give us. Some such graphics appear in the two excerpts.

Unfortunately, the cost to produce the book in color is prohibitive to pricing. To keep the price reasonable, the book is produced in grayscale. The readers, of course, can view each graphic in full color by running the provided R code. Alternatively, the reader can view any visualization in full color at webpage, https://analytics4strategy.com/book-look-inside.

For convenience, the herein two excerpts will present the color version.

2.2. What and Why of Access and "R"

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

The section will be an overview of the Access software and R. The discussion of "R" will be extended to introduce what the book calls layered charting. Written for self-directed learners, the remaining chapters will dive deeper into both software in the context of explaining by demonstration data-driven asset management practices made possible by them.

2.2.1. 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. 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 3 will go deeply into the how-to of super tables. Subsequent chapters will strengthen the readers skills with super tables through the explanations by demonstration of data-driven asset management practices.

Figure 2-5 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.

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.

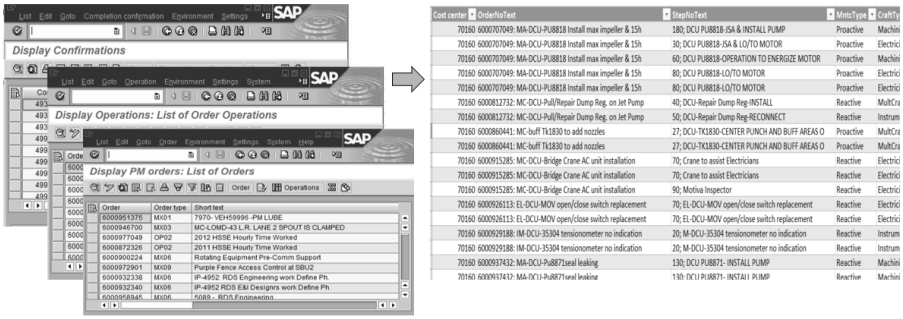

Figure 2-5: Three sub tables combined in a single super table.

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 limited 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 2-6.

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.

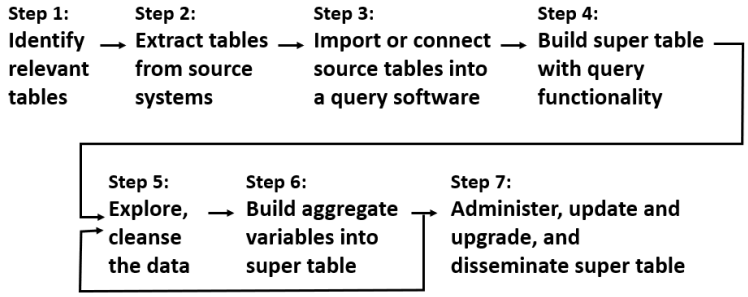

Figure 2-6: The process to build super tables.

Figure 2-7 shows the action to be taken. The tables of Figure 2-5 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.

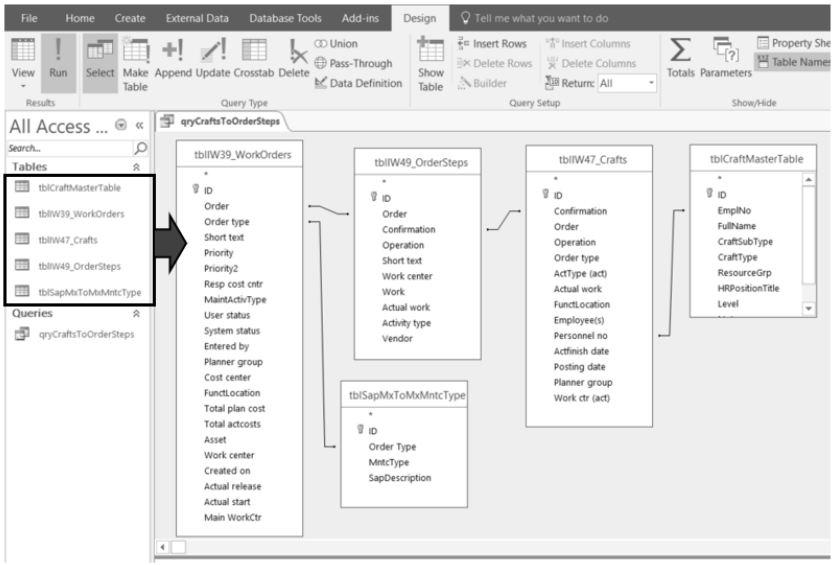

Figure 2-7: Tables imported to the query software and joined as a single table.

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 2-8. 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.

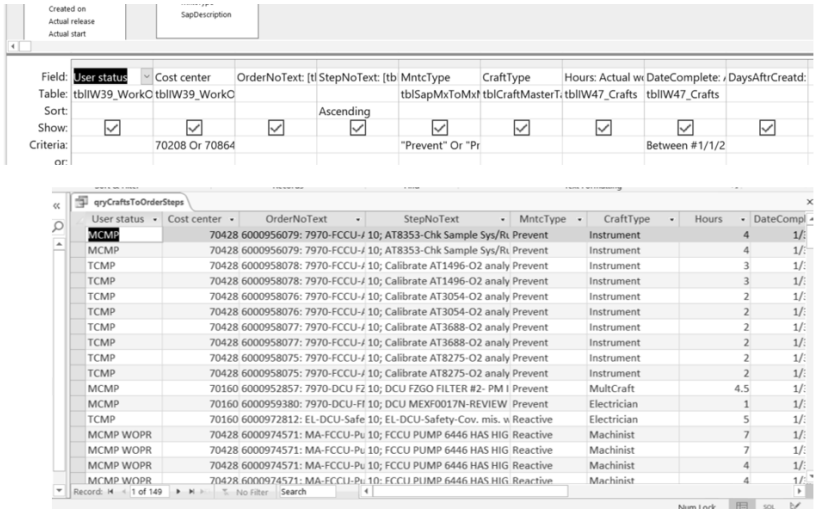

Figure 2-8: Variables pulled into the super table molded for insight.

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 txt 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.

2.2.2. "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 its 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 `lm` function 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 parallels 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-paralleled 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 4 will present and explain how to work with "R." The explanation of this chapter will be to give the reader the lay of the land. What is explained in the Chapter 4 will be extended in subsequent chapters as R is put in play for data-driven asset management practices.

Figure 2-9 show the primary three windows of "R." Like all software they can be moved and sized.

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 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.

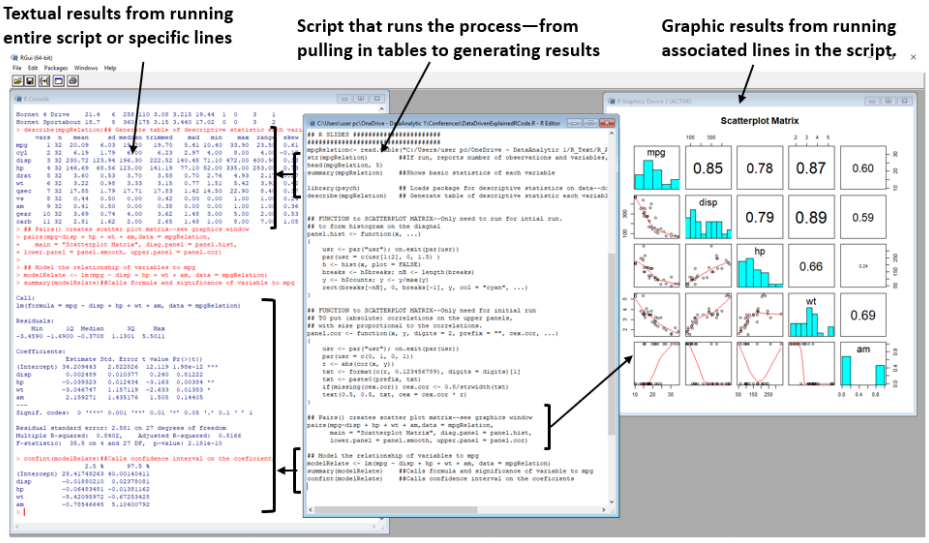

Figure 2-9: Console, script and graphic windows of R.

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 4.

2.2.3. 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.

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 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 2-10. Traditional and layered charting are shown side-by-side.

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 points. Layered charting offers many ways to regain granularity. Some are shown in the figure.

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

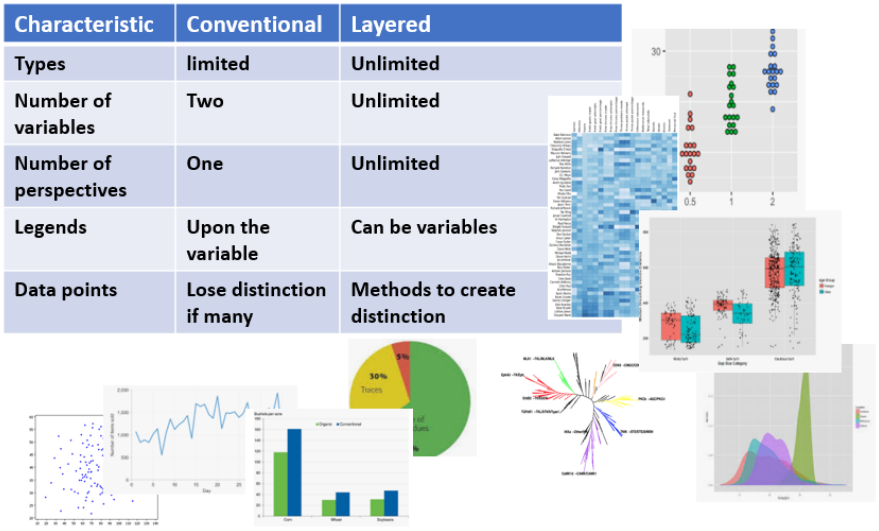

Figure 2-10: Contrasts between traditional and layered charting.

Figure 2-11 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.

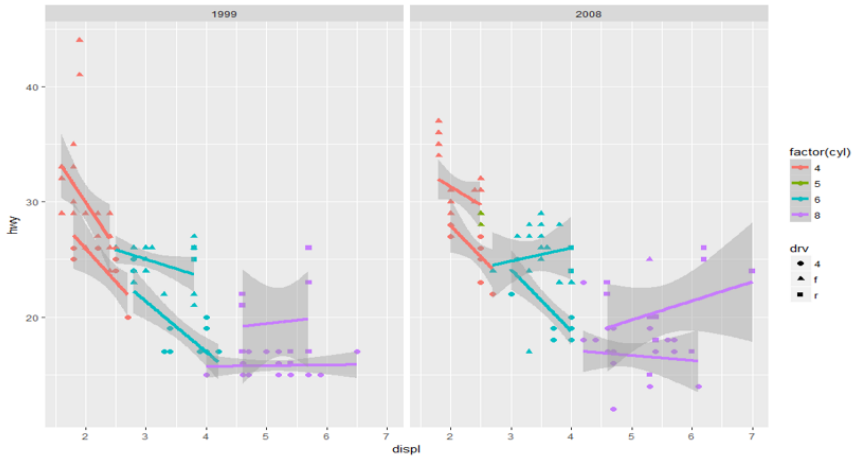

Figure 2-11: Layered charting to present measures of performance.

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.

Data and Analytics/Operational Effectiveness

Data and Analytics Skills for Your Career Security

Keeping it simple, only the skills you're likely to need

Richard G. Lamb

For those of us who are role holders in enterprise functioning, the personal purpose of acquiring practical working skills in data and analytics is to be able to better do what we already do and find new ways to do better yet. It follows that if you are a role holder who brings and incorporates data and analytics methods in your thoughts and tasks, your career outlook will be more secure and exciting. The book is written to be your gateway to the skills and to be the templates with which you will install the methods in your operational roles.

We all know that the field of data and analytics is huge and intimidating. It is a long slog to becoming comfortable. During the author's own long slog until arriving at the book, something exciting bubbled to the surface. There is a big difference between what we need to know and everything there is to know. We need to know what is possible as insight for decisions and functioning, we need to know how to get to the insight and, finally, we need to be able to interpret the insight. Just as the book does, we can leave the rest to the data scientists.

About the Author: In 2003, Richard Lamb, while struggling to get at the history captured in the databases of operational systems, found the skills to extract datasets of related history and join them in a super table of variables to make possible what was being envisioned for operational effectiveness. In 2014, Richard realized that, with statistical analytics and free enabling powerful pc-level software, an enterprise could ask and answer questions of operational effectiveness that are otherwise not possible. His activism to bring the epiphanies into the careers of role holders in the mainstream of operations has arrived at this book to explain data and analytics through the demonstration of methods.

Richard is a Registered Professional Engineer and Certified Public Accountant. He has previously authored two books: Availability Engineering and Management for Manufacturing Plant Performance, and Maintenance Reinvented for Business Performance. He has a BSCE, BBA and MBA from the University of Houston and a graduate-level Applied Statistics Certificate from the Texas A&M University.

https://analytics4strategy.com/data-and-code

ISBN 978-1-7343947-0-2

90000>

9 781734 394702

Analytics4Strategy