SESUG 2022 Paper 163

Data Wrangling in a Highly Regulated Industry

David B. Horvath, MS, CCP.

ABSTRACT

Data is rarely clean – it needs to be wrangled into a form that your analysts can actually use it. Wrangling that data becomes more complicated in a highly regulated industry such as Pharma or Banking/Finance. This session will look into those areas and provide suggestions on how to deal with the related issues.

INTRODUCTION

There are many different roles in the field of data analytics. The individuals who fill those roles have different skill sets or specialties. Some are great at creating models but not so good with dealing with the processes of getting the data. Others are great at the computer work getting the data but don't know much about statistics.

I have seen some brilliant models that make lousy use of the system resources in how they manipulate the data. Processes that take days to run, with proper resource usage, can run in hours. Often the tool is blamed: I hear a fair amount of "this ran long because SAS® sucks!" from Python-philes. And then turn around to see Python processes running extremely long. Not because of the tool but because of the way the code is written.

- While Quantitative Modelers (or Analysts) have the ability to develop code, their expertise lies more with the models and statistics.
- While Data Scientists have domain knowledge in the statistical and software development areas, their expertise lies more in the statistical.
- The Data Engineer has more expertise in the software development and data manipulation area.
- The Data Wrangler has strong expertise in developing software to process difficult forms of data
- While a person in each of these roles is expected to have some understanding and ability in the other areas, their specializations differ.

I am going to focus on Data Wrangling concerns along with others that affect those working in highly regulated industries. Many of these concerns affect those in all industries but with the regulators looking over our shoulders we have to carefully consider these issues.

DATA CONCERNS

The following areas are concerns in all organizations:

- Timeliness
- Provenance
- Correctness
- Availability

I am going to go over each of those areas in turn. For those of us within highly regulated industries, the regulators (or internal/external auditors before the regulators get there) can

shut you down. In smaller firms and industries with less scrutiny, it is possible to fall into bad practices in the name of "deadlines".

Let's start with a concrete example. Earlier this year, Equifax calculated scores incorrectly. Since I don't work with their data I can discuss it because my information is strictly publicly available. There are plenty of sources to read about this, https://www.forbes.com/advisor/personal-finance/equifax-credit-score-errors/ or https://www.cnn.com/2022/08/03/business/equifax-wrong-credit-scores/index.html are good places to start. They calculated scores incorrectly for a large number of customers over a period of several weeks. The score differences for many of the people was significant enough to increase the cost of borrowing (higher interest rate and fees) or even result in denials.

If you use that data, how do you deal with those errors? How do you even know if you are using that data? You may be using a credit bureau score – but do you know when bureau it is from?

TIMELINESS

Timeliness concerns primarily deal with the issues of when can we get it, how old is it when we do, and how frequently is it updated? We generally do not want to be working with old versions of data – unless we are actually looking back in time, then we want the data to be current to that historic date.

It really is up to your organization and application to answer these questions. In the highly regulated industries, we have to document those decisions and, often, document that we receive it in a timely manner.

Of course, the methods used to get the data is a concern. For me, automated approaches are preferred. After all, when there is a manual process, you need to have significant backups, documentation (hard to keep current), staff, and a procedure to handle the loss of the primary data-grabber.

I'm sure we've all see organizations where there is that one person who knows how to execute the really important processes. But that means they can't take a vacation when the data is due. But what happens if they win the lottery?

With lazy management, they'll let this happen because it is easier for them. Regulators really don't like seeing manual processes!

One of the important considerations with timeliness is the impact of using older data. While we want the latest data for the particular date, there may be situations where it is not available. By determining the impact of old data, you can then determine the response – hold downstream consumers or run the processes as normal.

I imagine the original data feed from Equifax was timely based on the normal processes. But what about the corrections?

The use of Shadow IT / Ghost Datasets can be scary because of the lack of automation and controls. This is compounded if they become sources of data to downstream processes.

Once Equifax corrects the data, especially if it is not part of the normal process, will the manual processes get the update? Will the Shadow IT / Ghost Datasets get fixed? In most examples I've seen, the answer is "not likely".

And what should your history represent? Should it be "fixed" (set to the correct values)? Or should it remain with the actual values received (so downstream can be audited)? Personally, a hybrid approach seems to be best – correction with retention of the actual (with plenty of notes)

PROVENANCE

Provenance is primarily concerned with the source and traceability of data. As it comes into the organization is the data from a trusted source and how can you be sure? Once it is in the organization there are often multiple versions of data as it flows through, then the question becomes: has it changed (and how can you be sure it has not)?

The complications increase the further into the organization the data is sourced. This is because each step it takes the more opportunity for changes.

This often becomes more of an issue where there is not strong Metadata Management or with diverse groups that don't have access to the metadata and data dictionaries. They each decide they need their own copy. Then someone discovers this copy, not knowing of the original or official location, makes yet another copy "to make sure it is available for their needs".

You will often see these in the form of "Shadow Datasets" – someone starts an extract or data store for their own purposes, others start using that data – because it is available. This is great until the creator stops building the spreadsheet. Or decides to change the layout (or selection criteria). Or when the creator has a bad day and makes mistakes on data entry (or the extract).

What happens is the data is not valid (or can't be processed correctly by the downstream system due to a layout change? There may be nobody to ask about changes.

With the Equifax situation, if we do not know the exact version of the data, we don't know if the score is actually any good!

CORRECTNESS

In the end, this all comes down to being sure the data correct and how we can be sure of that. If we are getting data from outside the organization, how do we determine if it is accurate?

If we are lucky, the source will tell us how the values are determined. But if they're willing to do that, then we could perform the calculations ourselves. For instance, if we buy property value data from Zillow ®, they are not going to provide us the calculations on how they determine those values.

But we can look at data trends. While the value of individual properties can vary widely – after a disaster for instance – in aggregate there should not be wide variance. In addition, the number of properties should not vary significantly.

Actually, the number of properties can vary significantly. The number of properties in the country do not change a lot but, since Zillow has to get data from individual county governments who do whatever they want. So, although this seems like a reasonable test of correctness, it is not.

In a similar vein, Fair Isaacs Company will not release the calculations for their score. In general, it is stable but variance is normal. It tends to go down during the Christmas shopping season and back up when tax refunds are issued.

The key is understanding the data and the way it behaves.

This is true with external data and that created within your organization. Controls are just as important internally as externally. This is a bit counter-intuitive because you know what kind of controls exist on systems within your organization. While you can trust, you need to verify.

With Equifax, you don't know how the score is calculated. But you can look for changes that do not match the normal patterns. If you detect questionable data then you have to decide

if it is better to use that data or use the prior version. Now, you could shut down your business (or the portion that uses that data) but that is not usually a possibility.

While there are other credit bureaus, the effort to switch is significant and the results don't match exactly. Not every creditor reports to every bureau, have different reporting cycles, and may perform score calculations on different days. While you could switch, that isn't going to happen quickly.

Whether the data provider is a sole source or not, if they go out of business, end the product line, or encounter a business discontinuity (disaster), you may not have access to it for a period of time. While you have the old values, how long can you safely continue to use them before the normal variance makes them invalid?

With manual data retrieval there are additional concerns. In particular, ensuring that the correct data was retrieved and the correct version of that data. You don't want to accidentally download the prior month once the current is available.

In the Equifax situation, you don't want to re-download the incorrect scores if corrected data is available.

Changing data – like Equifax corrections or restated economic data requires us to have a notification mechanism and ability to get the changes as quickly as they become available. And once the new data is received, verifying that it is, indeed, changed (rather than just a new file with the old data).

The issue that has pained me most personally is data providers lying about data formats and layouts. A few that I've encountered over the years include:

Unfortunately, data providers lie about data formats and layouts:

- CSV when fixed format promised (or vice versa)
- Zoned Decimal (mainframe sign "overpunch") when plain numeric promised (or vice versa)
- EBCDIC when ASCII promised (or vice versa)
- Incorrect record separators: UNIX when Windows promised (or vice versa)
- New fields added to middle of file rather than appended to end

Sadly, the "lies" don't end after getting into production. You need to be looking for these changes – regular data quality checks will help prevent incorrectly prepared data from impacting downstream systems.

AVAILABILITY

When we talk about availability we are primarily concerned with being able to get it every cycle (as defined by your data requirements). As mentioned under other categories, you do need to be prepared for the situation where the data is unavailable for a particular month or for the future.

These all exist no matter if the data source is automated or manual, internal or external, codified or shadow. It is particularly bad when the source silently ends – for instance, a manual / shadow database that is no longer updated. The data is available, but it is no longer timely – and it may not be directly obvious that it has not been updated (for instance, there is no last_update_date or the file name contains the date.

RESPONDING TO THESE CONCERNS

Unfortunately, there are no easy answers. The one advantage to being in a highly regulated industry means that we get support from auditors and regulators on preparing for issues and making the decisions on how to react.

Automation helps to standardize the processes month-over-month and documentation helps both understanding and continuity.

Many organizations will go to data vendors even when free versions of the information is available (economic data for instance). The vendor is expected to ensure the Timeliness, Provenance, Correctness, and Availability. There is usually a greater automated reception capability for the data (and subsequent provisioning) working with a vendor. The alternative is often grabbing the data from a government web page which is subject to change without notice.

The key is that a lot of the risk and responsibility is shifted to the Vendor – even if your organization is ultimately accountable for the results.

TOOL CONCERNS

Beyond the data, we have to worry about the tools we use. After all, there are many options for ETL and Analytics tools. There are essentially three main categories: Free Open source, Commercial Open Source, and Commercial. Free Open Source is something you download – operating systems like some Linux Distributions, Python tools, Python code packages, R tools, R code packages, etc. Commercial Open Source includes supported versions of all of these; often many of the components used are pure Open Source. And, of course, Commercial packages are those supported and maintained solely by a commercial entity.

No matter where you get your tools and components, auditors and regulators will ask similar questions about tools as data:

- Provenance – How do you know this is a good version?
- Correctness – How do you know the result is correct?
- Availability – What happens if it stops working (who do you contact)?

Coming up with acceptable answers are much easier with Commercial and Commercial Open Source than with Free Open Source.

Please do not think that I am anti-Open Source (Free or Commercial). My point is to show the additional risks you could face and concerns you should have.

For instance, I was asked how I could be sure a particular calculation was correct – the average/means of a particular field produced on a report. My response was "How do you know the average is correct in excel? Because Microsoft tells you so. How do I know it is correct in TOOL? Because VENDOR tells me so." In that specific case, the tool/vendor was SAS. If it was Jupyter Hub, I don't have that promise.

With a reputable company you have support. It doesn't matter if you're dealing with Commercial or Commercial Open Source products: using a tool like SAS or Anaconda (Python and R distribution) you have some support. You have someone standing behind quality. You can even arrange to have a contract where source code is held in escrow.

If SAS, Inc. goes out of business, with the code in escrow, we could update it ourselves as we migrate to different operating systems. Now, I don't expect for them to do so, but it makes regulators a lot happier to know that we could.

When it comes to tools and components that are Free Open Source, it becomes much harder to answer the questions about Provenance, Correctness, and Availability. A consideration is how you'd handle the situation where the tool goes away or loses support – this is of particular concern with Python packages and the like.

I have some recent examples to share. I am not a "SAS bigot" and have used both Commercial and Free Open source. It is a matter of risk…

LOSS

The Open Source code you are using could disappear in the blink of an eye. In 2016, one developer withdrew 250 modules from the JPM repository in a moment of pique. One of his modules was named "Kik" which is also the name of an instant messaging application. Kik's lawyers saw the name, sent a cease and desist requesting a rename to the developer who refused, then the lawyers went to the repository who pulled that one module. One of those 250 modules was one, left-pad, that was very popular. As a result, the thousands of applications that used that function failed.

Imagine going to your CEO or a regulator and having to explain why everything is shut down?

Fortunately for the users of that module, the repository restored it and another developer has taken over support. It could've been much worse if the original developer decided to release a new version that purposely did not work!

How is your organization impacted if this happens?

The reason that the tool was withdrawn really doesn't matter. The important fact is that one of the components your organization relies on to exist does not. There are many different reasons this could occur:

- Annoyance at unrelated organization's actions (this example)
- Inclusion of Employer's IP
- Inclusion of Other's IP (like a patent or copyright violation)
- Targeting Your organization for some reason

Error! Reference source not found. is a screen print of a news article about a developer removing code from a repository.

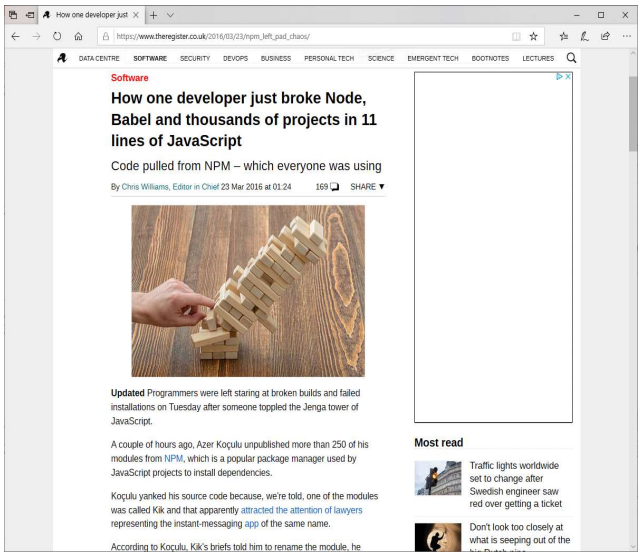


Figure 1. Screenprint of News Article: Developer Removes Code from Repository, Hilarity Ensues.

https://www.theregister.co.uk/2016/03/23/npm_left_pad_chaos/

LOSS OF SUPPORT

In some ways, the loss of support of a tool can be even worse because it is hidden. Suddenly, there just are not more changes. If the tool continues to work, you won't even notice that it happened.

With Commercial or Commercial Open Source, at least there is a vendor to provide the support (and should notify you when a tool is going to sunset). But many pure Free Open Source tools were created purely for fun or to solve a personal problem. That means ongoing support is also "for fun" (as opposed as "for compensation"). At some point, interest will wane.

An interesting example of ongoing support is the AWK programming language. For those not familiar, it is a scripting language created by Aho, Weinberger, and Kernighan in 1977 with strong roots in C (for obvious reasons). A recent (August 2022) article describes how Brian Kernighan is still maintaining and improving the language.

Fortunately, AWK is included in the core of UNIX/Linux distributions so is also being maintained by the vendors that sell those products.

https://blog.adafruit.com/2022/08/24/unix-legend-brian-kernighan-still-maintaining-awk-code-unix/

Another example of loss of support is the Stratux tool. The Stratux project is a software installation that runs on Raspberry Pi to extract broadcasted aircraft position data (called ADS-B) and serve it up to the Electronic Flight Books used by private pilots. It is also used by plane spotters. At some point in 2021 or 2022, CYoung got tired of maintaining the application. Fortunately, there were others willing to take this on and add new functionality. Otherwise, at some point, it would not be usable – it does not work on newer Raspberry Pi versions (3B+ for instance), so if your current unit failed and only newer were available, you'd lose your receiver. As of the time I'm writing this, the primary web site http://www.stratux.me is still offline although the github has been taken over.

If you are using a tool (any tool), eventually there will need for changes to be made because of:

- Changes in versions/libraries (like Python 2.x to 3.x)
- Changes in environment (negative interest rates for instance)
- Changes in operating system/version
- Changes in hardware (CPU to GPU for instance)
- Or even general maintenance when incorrect results are discovered

ORGANIZATIONAL RESPONSE TO LOSS/LOSS OF SUPPORT

So what does your organization do if the code is lost or need support?

- Can you work from a copy of the original code? There may be legal reasons you can no longer use it.
- Is the code even readable? Obfuscation is common or you may not be able to track down a complete copy
- Does your team have the technical skills to maintain it? If you are a Python shop and the code is Python, you are in a much better position than if the code is written in C.
- Does your team have the domain knowledge? While it might be easy to write a new left-pad function, creating a ML package would be a bit harder!
- And even if your team has the skills and knowledge, do you have the time? What else will need to be deferred as a result?

After all, the reason for an external tool or package is so that your team does not have to expend the time or effort to create that code.

In addition to the tools you know about, there is a danger that you may have hidden tools. I had a consultant try to implement a solution that used a downloaded package. This was in an environment where we aren't allowed to "just download packages". When I expressed concern, the response was "It is scripting, you can always maintain it."

Oh, great. If I had time to be an expert in processing this kind of data, I would've written it myself and not hire you.

Now, personally, I could update and recompile the Stratux code. I've already been doing some "strange things" to save off some of the received data. But if I did not have that ability and another enthusiast did not take it on, I'd lose a tool that is important to my situational awareness when I'm in the front seat.

MALWARE

The situation can be different if the tool/code you are using includes malware. While there may be monitoring of code in packages, when obfuscated, it may not be obvious. There is limited use of Python and R in malware scanning products.

Overt forms of malware could be particularly damaging because they will be obvious pretty quickly because they delete or encrypt your data (ransomeware).

But subtle forms can be even worse. One suggested form changes the results in certain ways after an extended elapsed time, number of records processed, every Nth time executed, or after a certain number of times executed. That way, your code and model validation processes succeed and then stop working in strange ways. In addition, data could be exfiltrated out of your organization if your firewalls allow access to the Internet.

Error! Reference source not found. is a screen print of a news article about malware on GitHub.

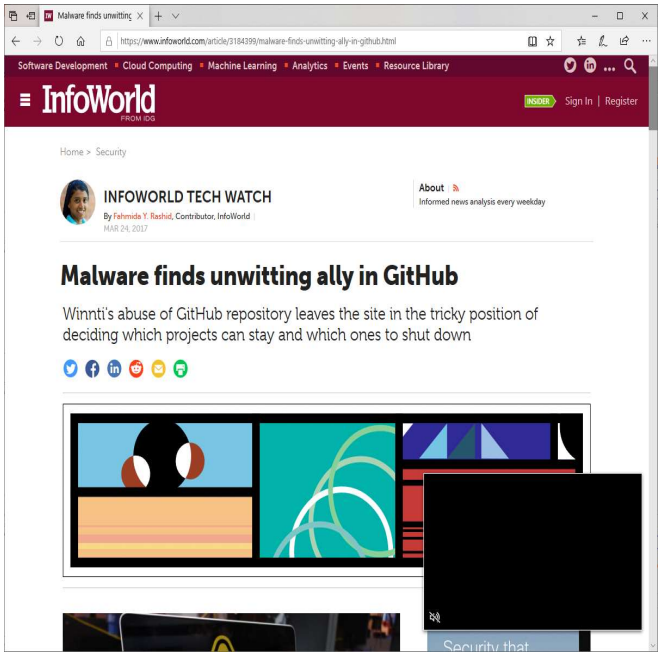


Figure 2. Screenprint of News Article: Malware on GitHub.

https://www.infoworld.com/article/3184399/malware-finds-unwitting-ally-in-github.html

TARGETED

You could find yourself in a difficult position if the developer of the tool or package has a reason to dislike your application or organization. For instance, the employees of GitHub tried to get the company to kick US Immigrations and Customs Enforcement ("ICE") off the platform.

Error! Reference source not found. is a screen print of a news article about GitHub employees asking the company to stop supporting ICE.

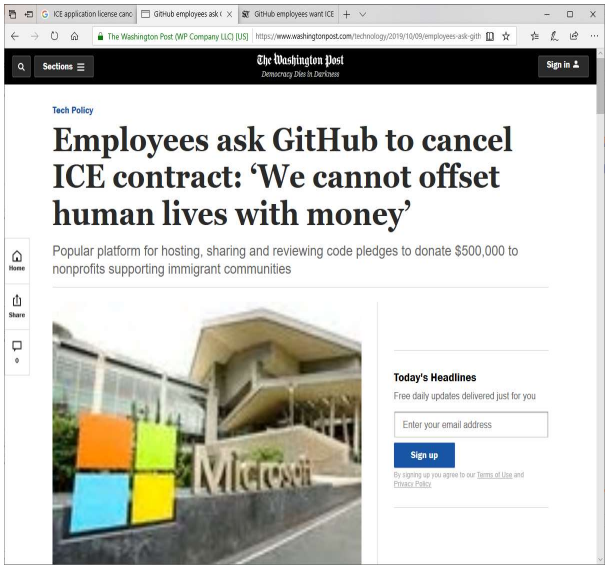


Figure 3. Screenprint of News Article: GitHub Employees asking Company to Drop ICE as Customer.

https://www.washingtonpost.com/technology/2019/10/09/employees-ask-github-cancel-ice-contract-we-cannot-offset-human-lives-with-money/

Fortunately for ICE, there was a commercial contract. But in the absence of a contract, the Terms of Service apply which pretty much reserve all rights to the company. Take a look at the Facebook Terms and Conditions. If they decide to cut you off, you'll lose access to years' worth of your pictures, videos, posts, and conversations. If GitHub cut you off, that would leave your data or code stranded and inaccessible. What would the impact be then?

Pick any other tool your organization uses; if you don't have a contract, how do you force access to your data or usage of that tool?

This is especially true if the creator decides to enforce their decision:

- Through forced failures – by modifying their code to target you preventing your use
- Through legal means – going to court with claims of intellectual property violations
- Through overt disabling – a version of code that prevents it from running just for you
- Through covert means – a version of code that subtly changes behavior just for you

When you are talking about Open Source Software, do you have any rights to correct results? While most Free and Commercial Open Source Software is released under licenses that are very liberal, nothing prevents a developer from including exclusions. "This tool is released under terms of the BSD license except that it may not be used by any organization that profits from the extraction of fossil fuels." Just to be clear, that exclusion was not my creation.

If packages are introduced into your organization without robust vendor management (including license review) and code reviews, something like this could occur. Then you are faced with code that is critical to your business that you actually aren't allowed to use.

Even with code reviews, if the code is obfuscated, you won't know if there are targeted triggers.

MITIGATION

Well, to mitigate the risks, stick with Commercial and Commercially Supported Open Source Software. As a result you will give up some speed of innovation. Or you can use Free Open Source but with tight controls. Consider limiting the use of these in business critical applications.

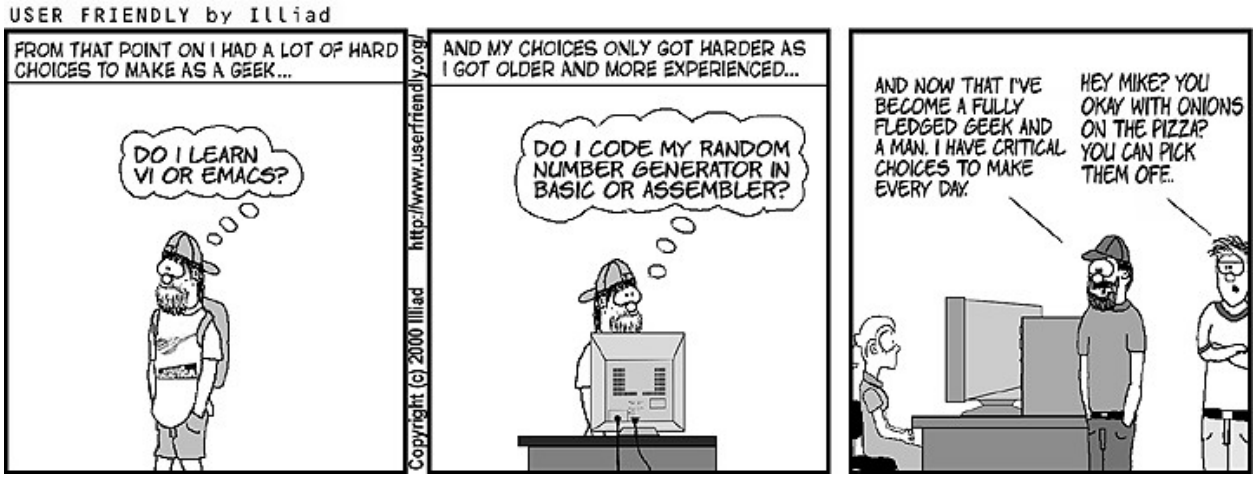
While not perfect, going to a vendor reduces risk of the other issues and provide mitigation:

- Recreating applications that may contain IP
- Taking on support of unsupported code
- Scanning for malware

CONCLUSION

It is all about choices. Do you only use data and tools from commercial sources or will you use Free/Open sources?

- Sometimes the risk is worth the gain
- Sometimes not



ACKNOWLEDGMENTS

I would like to thank the organizers of this conference. This is neither my first SESUG nor first time speaking at SESUG – I've always enjoyed working with the organizers.

I also want to thank my employer for their support in attending (and speaking at) this conference even though I'm not allowed to mention their name. Last, and certainly not least, is my spouse Mary who doesn't complain when I spend time working on conference materials.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

David B. Horvath, MS, CCP
+1-610-859-8826
Email: dhorvath@cobs.com
Web: http://www.cobs.com
LI: http://www.linkedin.com/in/dbhorvath