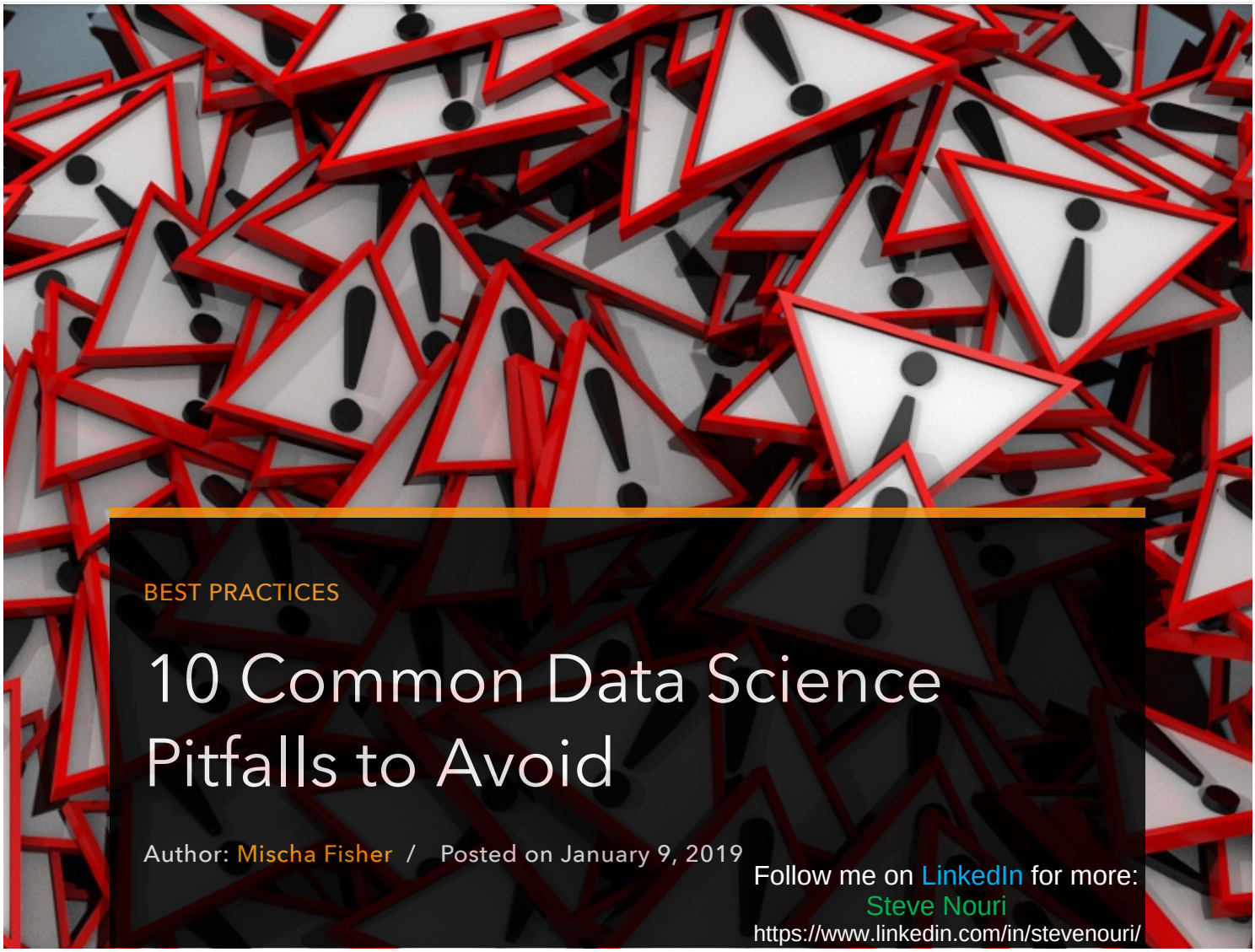

The use of artificial intelligence tools and techniques is rapidly expanding in businesses. However, frequently overshadowed by the major breakthroughs are the pitfalls that every individual analyst, data scientist, team, and enterprise should guard against.

My current role in overseeing several Illinois state agencies—each with their own unique set of quantitative challenges and unanswered questions—does not necessarily create a lot of methodological overlap. However, one commonality across all activities in our 60,000 person enterprise is avoiding the analytic pitfalls

that are universal to every data practice. Below is a list of the ten common data science pitfalls we do our best to avoid.

1. Making Unrealistic Assumptions

It's simple to say 'garbage-in, garbage-out,' but it's also regrettably common for analytical products that follow all the right steps and draw the correct conclusions from the outcomes to start from a place of misguided underlying assumptions. This is particularly true in public sector situations where politics is an inescapable reality; however, dogma and incorrect preconceptions can easily find themselves embedded in analysis within the private sector as well. My favorite example we've directly dealt with are profitability projections for high speed rail projects that started with an underlying assumption that construction costs per mile within the US would track relatively closely to what was managed in Taiwan, or that ridership level inputs to the model in an area with dense auto ownership would mirror that in an area with low density ownership. The math was correct, but the inputs were most certainly based off of incorrect assumptions.

Scenario	Capital Available	Funding Gap	Construction Cost	Percent Financeable
CHI to STL, Low CAPEX, Baseline Revenue, Simple DCF	$1,852	$21,148	$23,000	8%
CHI to STL, Low CAPEX, Baseline Revenue, CAB only	$2,762	$20,238	$23,000	12%
CHI to STL, Low CAPEX, Baseline Revenue, CAB/CI mix	$2,549	$20,451	$23,000	11%
Full System, High CAPEX, Baseline Revenue, Simple DCF	$3,331	$46,669	$50,000	7%
Full System, High CAPEX, Baseline Revenue, CAB only	$5,088	$44,912	$50,000	10%
Full System, High CAPEX, Baseline Revenue, CAB/CI mix	$4,694	$45,306	$50,000	9%
Full System, Low CAPEX, Optimistic Revenue, Simple DCF	$4,343	$25,657	$30,000	14%
Full System, Low CAPEX, Optimistic Revenue, CAB only	$6,800	$23,200	$30,000	23%
Full System, Low CAPEX, Optimistic Revenue, CAB/CI mix	$6,249	$23,751	$30,000	21%

Figure 1: Scenario estimates for high speed rail profitability (Source: 2013-09-24 - 220 MPH High Speed Rail Preliminary Study, UIUC & UIC for IDOT, pg. 33)

2. Measuring The Wrong Thing

Like unrealistic assumptions, this can doom the outcome of your project before you've tested a single model or written a single line of code. The most common way to do this is modeling to assess the quality of programmatic or business initiatives that judge overall effectiveness by inadvertently measuring an input as a form of output. A great example of this—highlighted by Stanford economist Thomas Sowell—is the tendency for indices that measure the quality of universities to use inputs to the university production function (payrolls, library sizes, etc.) as a measure of quality. While these are certainly proxies for overall university funding, they are not in fact measurements of the output, where a better gauge of quality is instead the effect the institution has on students passing through it controlling for all other factors.

```
universities <- data.frame(
 name = c("Harvard", "UIUC", "Yale", "UC Berkeley", "Columbia University", "UM Ann A
     "UT Austin", "University of Chicago", "UCLA", "IU Bloomington", "Stanford", "UW
     "Cornell", "Princeton", "UW Seattle", "UM Twin Cities","UNC Chapel Hill",
     "University of Pennsylvania", "Duke", "Ohio State Columbus"),
 books = c(16.8, 13.2, 12.8, 11.5, 11.2, 10.8, 10.0, 9.8, 9.1, 8.6, 8.5, 8.4, 8.1, 7
 ranking = c(2, 46, 3, 22, 3, 27, 49, 3, 19, 89, 7, 49, 16, 1, 59, 76, 30, 8, 8, 56)
)

library(ggplot2)
library(ggrepel)
ggplot(universities, aes(books, ranking)) + geom_point() + geom_smooth(method='lm',
 geom_text_repel(aes(label=name)) + xlab("Books in Millions") + ylab("Rank") +
 scale_y_reverse( lim=c(100,0))
```

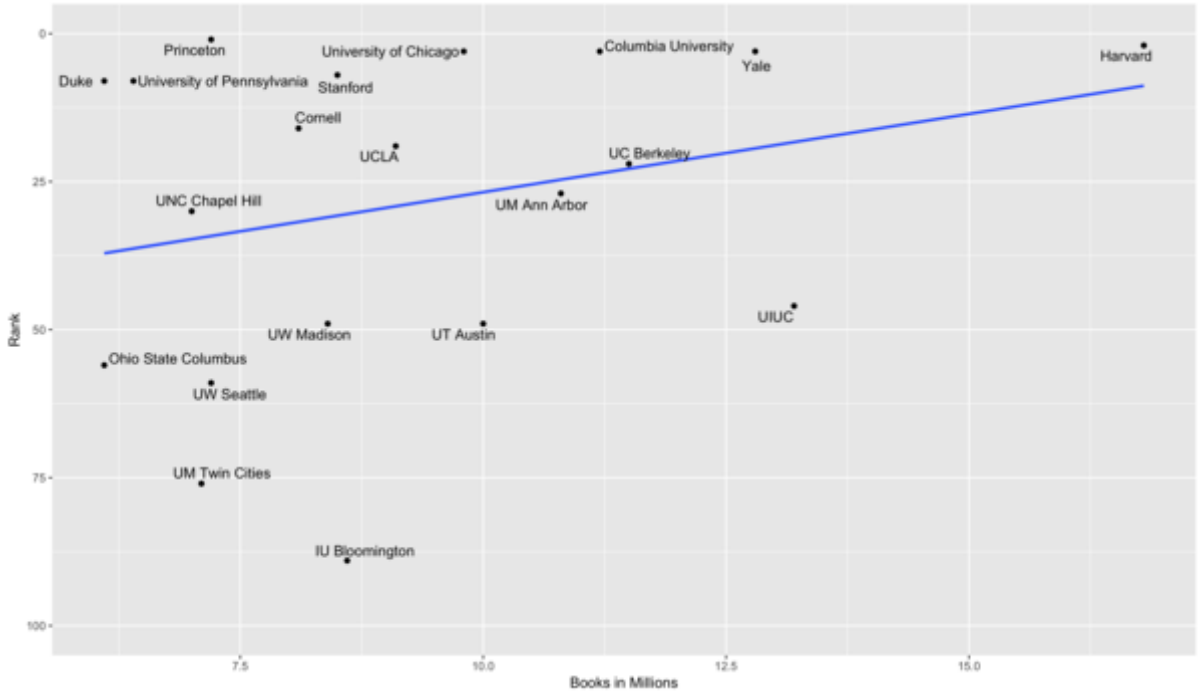

Figure 2: Association Between a Popular Ranking of University Quality and Their Library Sizes

This principle can be applied in dramatically different areas. For instance, perhaps your organization runs an automated ML system that distributes content to clients via email based on connected device behavior, and you're measuring the effectiveness of the tool based on the association between user behavior and the number of outgoing emails the system produces, rather than the performance of the algorithm by whatever the end-outcome is that those emails are intended to accomplish. Or similarly, you might find yourself measuring user time spent on an app because you're assuming it will impact client satisfaction, but if you inadvertently make the app worse user time will increase on a per-task basis, distorting your understanding of client satisfaction because you weren't measuring the right thing to begin with.

3. Spurious Correlation Due To Poorly Constructed Questions

This falls under the category of finding a strong relationship between variables but failing to account for some sort of logical or theoretical link between what you're inferring. Traditionally, this is thought of in the context of relatively simple cross-tabulations or regressions and we have to remind ourselves that there should be a logical link between two correlational trends. Tyler Vigen maintains a somewhat infamous website with a myriad of great examples, including the below association between per capita margarine consumption and divorce rates in Maine.

```
## Spurious Correlation http://tylervigen.com/spurious-correlations ###

library(tidyr)

spurious <- data.frame(
 year = c(seq(from=2000, to=2009)),
 per_capita_margarine_consumption_in_lb = c(8.2, 7, 6.5,5.3, 5.2, 4, 4.6, 4.5, 4.2,3
divorce_rate_in_maine_per_1000 = c(5, 4.7, 4.6, 4.4, 4.3, 4.1, 4.2, 4.2, 4.2, 4.1)
)
cor.test(spurious$per_capita_margarine_consumption_in_lb, spurious$divorce_rate_in_m

spurious$diff_marge <- spurious$per_capita_margarine_consumption_in_lb / lag(spuriou
spurious$diff_div <- spurious$divorce_rate_in_maine_per_1000 / lag(spurious$divorce_

spurious <- select(spurious, year, diff_marge, diff_div)
spurious <- gather(spurious, key=year)
spurious$series <- spurious$year
spurious$year <- rep(seq(2000, 2009,length.out=10),2)


ggplot(spurious, aes(year, value, color=series)) +
 geom_point() + geom_line(size=2) + ylab("Change") + xlab("Year") +
 scale_x_continuous(breaks=c(2000, 2002,2004,2006,2008,2010)) + ggtitle(".9925 Pears
```

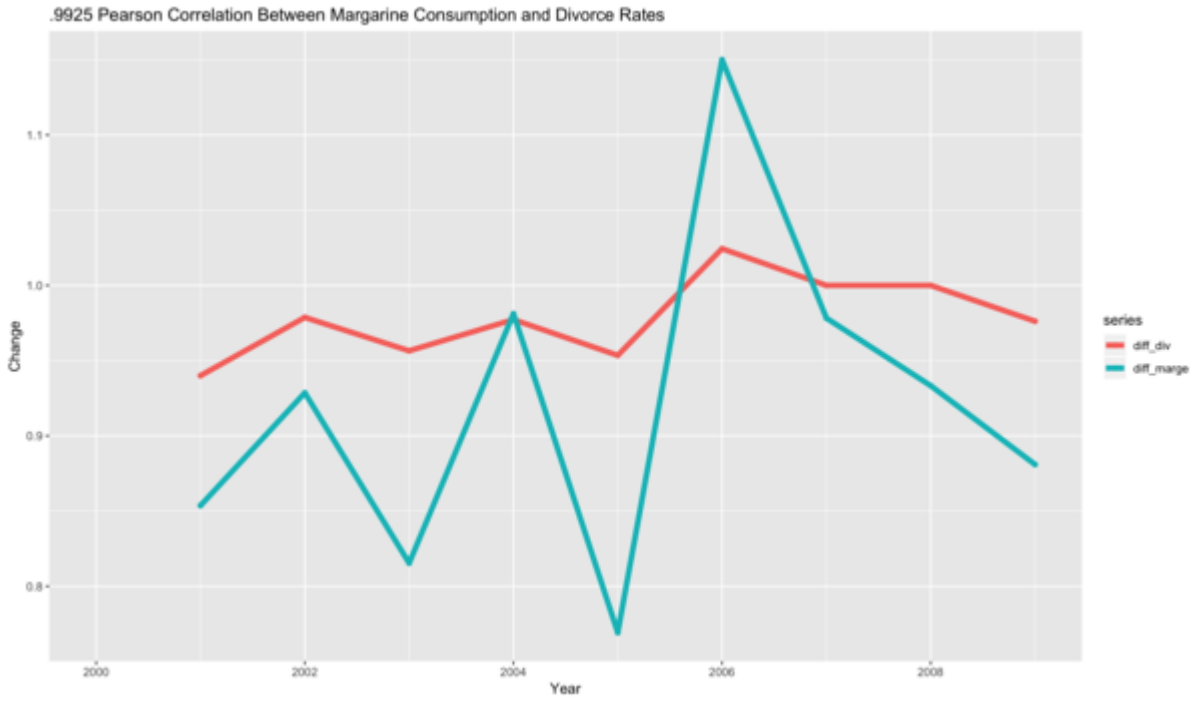

Figure 3: Spurious Correlation Between Divorce Rates in Maine and Per Capita Margarine Consumption

Therefore, before beginning any examination of factors one should consider whether or not 1) there is a linkage or some other confounding factor that you're missing, or 2) there is a pre-existing trend unlinked to any factor being studied but that coincidentally your variables are all following (such as growing at some percent over a period of time). Newer tools and computational power brought about by deep learning methods both mitigate and exacerbate new challenges in correlational examinations for mathematical reasons (see the curse of dimensionality) as well as the fact that since there may be nth order relationships being sought, the coefficients can be hard to interpret or act upon and the computation is largely a black box for potential decision makers.

4. Spurious Correlation As A Result Of Bad Math

Unlike the example above where the failure is one of theory or model construction, spurious correlation can also result by forgetting how your inputs were constructed. This can be particularly common when looking at relationships among certain performance indicators; if your organization uses KPIs on a per user or per customer basis, you might inadvertently map the relationship between two KPIs that are mathematically derived quotients and find a relationship that's really the mapping of their common divisor.

```
library(ggplot2)

## Multiplot Function courtesy of http://www.cookbook-r.com/Graphs/Multiple_graphs_c
## Modified Multiple plot function with title courtesy http://www.guru-gis.net/multi

multiplot <- function(..., plotlist = NULL, file, cols = 1, layout = NULL, title="",
           fontsize = 12, fontfamily = "Helvetica") {
 require(grid)

 # Make a list from the ... arguments and plotlist
 plots <- c(list(...), plotlist)

 numPlots = length(plots)

 # If layout is NULL, then use 'cols' to determine layout
 if (is.null(layout)) {
  # Make the panel
  # ncol: Number of columns of plots
  # nrow: Number of rows needed, calculated from # of cols
  layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
          ncol = cols, nrow = ceiling(numPlots/cols))
 }

 if (nchar(title)>0){
  layout <- rbind(rep(0, ncol(layout)), layout)
 }

 if (numPlots==1) {
  print(plots[[1]])

 } else {
  # Set up the page
  grid.newpage()
```

```
    pushViewport(viewport(layout = grid.layout(nrow(layout),
                         ncol(layout),
                         heights = if (nchar(title)>0) {unit(c(0.5, rep(5,nrow(layout)
                         else {unit(c(rep(5, nrow(layout))), "null")})))

    # Make each plot, in the correct location
    if (nchar(title)>0) {
     grid.text(title,
          vp = viewport(layout.pos.row = 1, layout.pos.col = 1:ncol(layout)),
          gp = gpar(fontsize = fontsize, fontfamily = fontfamily))
    }

    for (i in 1:numPlots) {
     # Get the i,j matrix positions of the regions that contain this subplot
     matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

     print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                     layout.pos.col = matchidx$col))
    }
   }
  }

  spurious <- data.frame(
   TotalSalesByRegion = rnorm(200, mean = 100000,sd=3000),
   TotalMarketingBudgetByRegion = rnorm(200, mean = 10000, sd=300),
   EmployeesPerRegionalTeam = sample(5:20, 200, replace=T)
   )


  p1 <- ggplot(spurious, aes(TotalMarketingBudgetByRegion, TotalSalesByRegion)) +
   geom_point() + geom_smooth(method='lm')
  p2 <- ggplot(spurious, aes(TotalMarketingBudgetByRegion, EmployeesPerRegionalTeam))
   geom_point() + geom_smooth(method='lm')
  p3 <- ggplot(spurious, aes(EmployeesPerRegionalTeam, TotalSalesByRegion)) +
   geom_point() + geom_smooth(method='lm')

  spurious$KPISalesPerEmployee<- spurious$TotalSalesByRegion/spurious$EmployeesPerRegi
  spurious$KPIMarketingPerEmployee <- spurious$TotalMarketingBudgetByRegion/spurious$E
  p4 <- ggplot(spurious, aes(KPIMarketingPerEmployee, KPISalesPerEmployee)) +
   geom_point() + geom_smooth(method='lm')

  multiplot(p1, p2, p3, p4, cols=2)
```

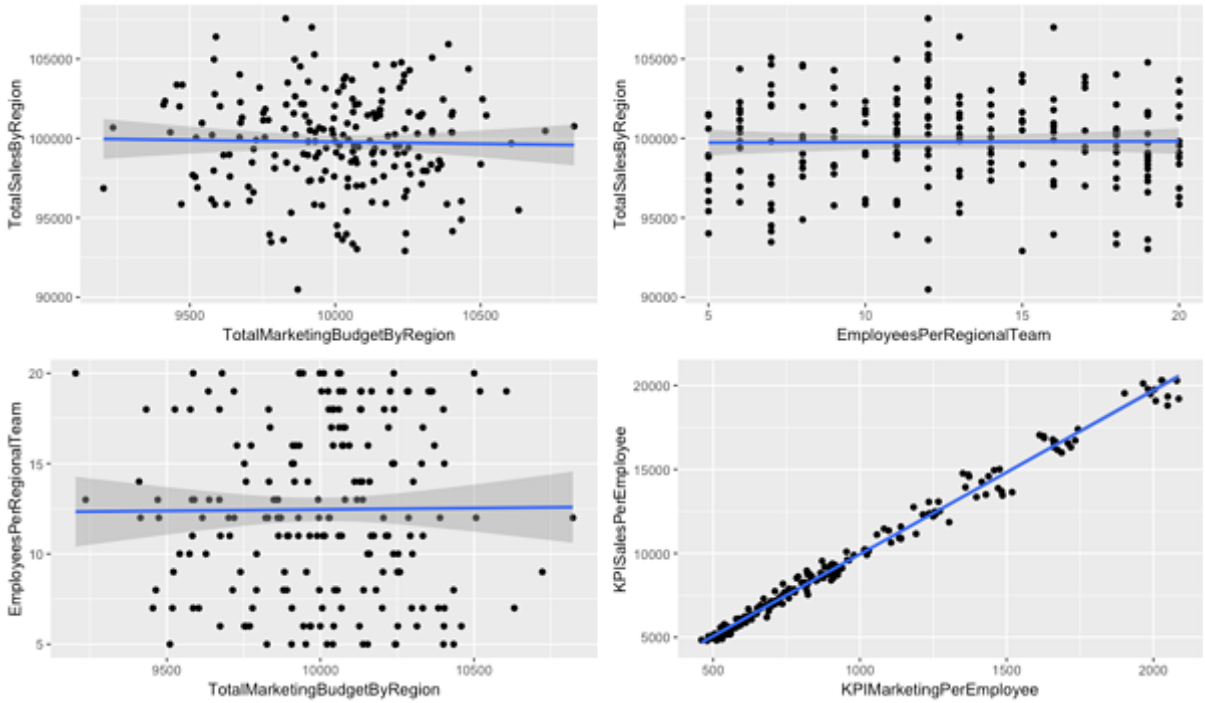

Figure 4: Spurious Correlation Among Unrelated KPIs Due to Bad Math

Perhaps, as in the above example, you're looking at regional sales teams and your organization has KPIs that include average sales per team member and average marketing budget per team member across each regional team. Examining the relationship between these variables would find a strong relationship between the sales expenditures and marketing budget, when your regression slope is really tracking the common divisor (number of team members) in the quotient.

5. Failing To Consider Grouping And Clustering Effects (Simpson's Paradox)

This is in some ways a form of spurious correlation, but it is specific and novel enough that it deserves its own mention. For the uninitiated, the paradox can be summarized by noting that one can find a strong statistical relationship between

two variables, without any of the previously listed errors—there can be a satisfactory theoretical logic linking them, correctly constructed variables, accurate measurements, and no bad assumptions. Still, after clustering the data separately on some other attribute or set of attributes, the direction of the correlation can be totally reversed.

```
simpsons1 <- data.frame(
 p1 = rnorm(1000, mean=15000, sd=3000),
 p2 = seq(20000, 10000,length.out = 1000)
)
simpsons1$p2 <- simpsons1$p2 - 0.5 * simpsons1$p1

simpsons2 <- data.frame(
 p1 = rnorm(1000, mean=10000, sd=2000),
 p2 = seq(12500, 7500,length.out = 1000)
)
simpsons2$p2 <- simpsons2$p2 - 0.5 * simpsons2$p1

simpsons3 <- data.frame(
 p1 = rnorm(1000, mean=5000, sd=2000),
 p2 = seq(9000, 5000,length.out = 1000)
)
simpsons3$p2 <- simpsons3$p2 - 0.5 * simpsons3$p1

simpsons4 <- data.frame(
 p1 = rnorm(1000, mean=3500, sd=1500),
 p2 = seq(6000, 3000,length.out = 1000)
)
simpsons4$p2 <- simpsons4$p2 - 0.5 * simpsons4$p1

simpsons5 <- data.frame(
 p1 = rnorm(1000, mean=1500, sd=1000),
 p2 = seq(3000, 1500,length.out = 1000)
)
simpsons5$p2 <- simpsons5$p2 - 0.5 * simpsons5$p1


simpsons <- rbind(simpsons1, simpsons2, simpsons3, simpsons4, simpsons5 )
simpsons$group <- c(rep("Group 1", 1000), rep("Group 2", 1000), rep("Group 3", 1000)


p1 <- ggplot(simpsons, aes(p1, p2)) + geom_point(alpha=0.5) + geom_smooth(method='lm
```

```
p2 <- ggplot(simpsons, aes(p1, p2, color=group)) + geom_point(alpha=0.5) + geom_smoo

multiplot(p1, p2, cols=2)
```

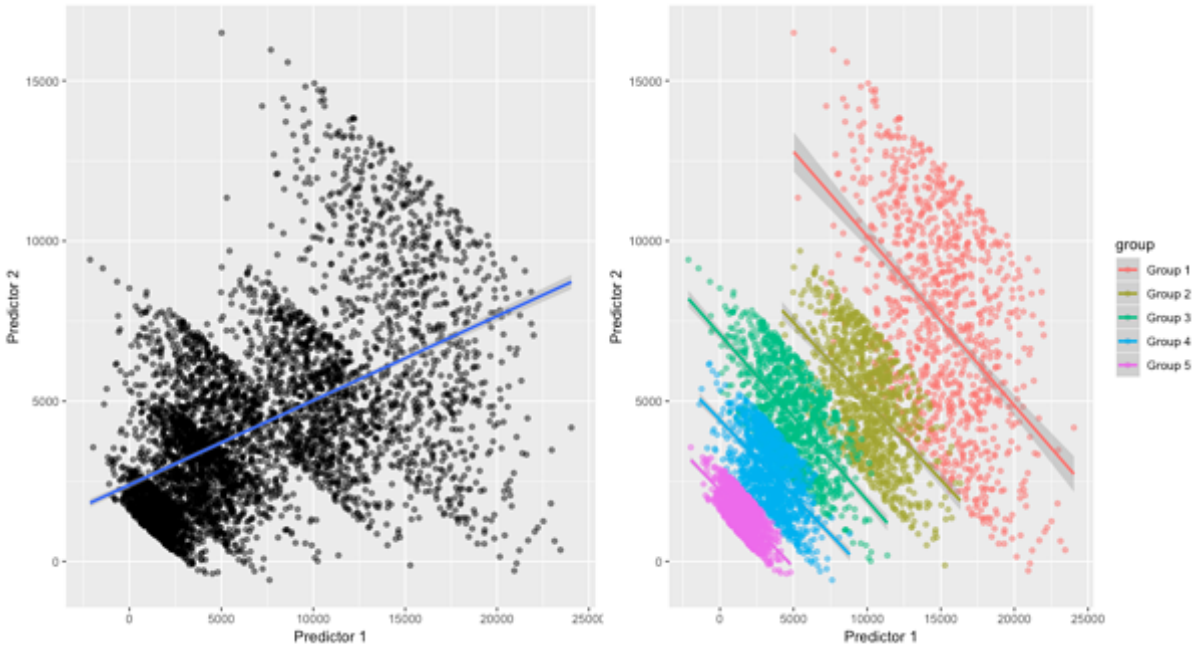

Figure 5: Clustering and Simpson's Paradox

This phenomenon has been chronicled for over 100 years and earned its name from Edward Simpson in 1951. There are numerous academic discussions of it within the statistical literature that estimate various probabilities of the paradox being present in a dataset, and statistical packages have been developed to test for it by increasing the probability of finding previously unseen categories.

While new tools may make it easier to discover incidences of the paradox at play, at its core the problem is a failure of considering potential clustering effects when mapping potential relationships.

6. Forgetting About Distribution Theory

A lot of analysis focuses on mean differences across groups, controlling variance, or examining correlations (as noted in the examples above). However, it's critical to remember that these are moments of a distribution, not the distribution itself. And even though moments like the mean, variance, skewness, and kurtosis can all be useful, they are not by themselves substitutes for understanding the characteristic function or simply visualizing the data you're examining. Failing to recognize that vastly different distributions can produce similar summary statistics can lead to inferential errors or faulty assumptions anywhere along your analytical workflow.

```
lanscombe <- data.frame(
x1 = c(10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5),
y1 = c(8.04, 6.95, 7.58, 8.81, 8.33, 9.96, 7.24, 4.26, 10.84, 4.82, 5.68),
y2 = c(9.14, 8.14, 8.74, 8.77, 9.26, 8.10, 6.13, 3.10, 9.13, 7.26, 4.74),
y3 = c(7.46, 6.77, 12.74, 7.11, 7.81, 8.84, 6.08, 5.39, 8.15, 6.42, 5.73),
x2 = c(8, 8, 8, 8, 8, 8, 8, 19, 8, 8, 8),
y4 = c(6.58, 5.76, 7.71, 8.84, 8.47, 7.04, 5.25, 12.50, 5.56, 7.91, 6.89)
)

p1 <- ggplot(lanscombe, aes(x1,y1)) + geom_point(size=2) + geom_smooth(method='lm',
 ylim(0,20) + xlim(0,20) +
 geom_vline(xintercept=mean(lanscombe$x1), color='red')
p2 <- ggplot(lanscombe, aes(x1,y2)) + geom_point(size=2) + geom_smooth(method='lm',
 geom_hline(yintercept=mean(lanscombe$y2), color='red') + ylim(0,20) + xlim(0,20) +
 geom_vline(xintercept=mean(lanscombe$x1), color='red')
p3 <- ggplot(lanscombe, aes(x1,y3)) + geom_point(size=2) + geom_smooth(method='lm',
 geom_hline(yintercept=mean(lanscombe$y3), color='red') + ylim(0,20) + xlim(0,20) +
 geom_vline(xintercept=mean(lanscombe$x1), color='red')
p4 <- ggplot(lanscombe, aes(x2,y4)) + geom_point(size=2) + geom_smooth(method='lm',
  geom_hline(yintercept=mean(lanscombe$y4), color='red') + ylim(0,20) + xlim(0,20) +
  geom_vline(xintercept=mean(lanscombe$x2), color='red')

multiplot(p1, p2, p3, p4, cols=2, title = "X1:X2 μ = 9 σ² = 11, Y1:Y4 μ = 7.5, σ² =

sapply(lanscombe, mean)
sapply(lanscombe, var)
```

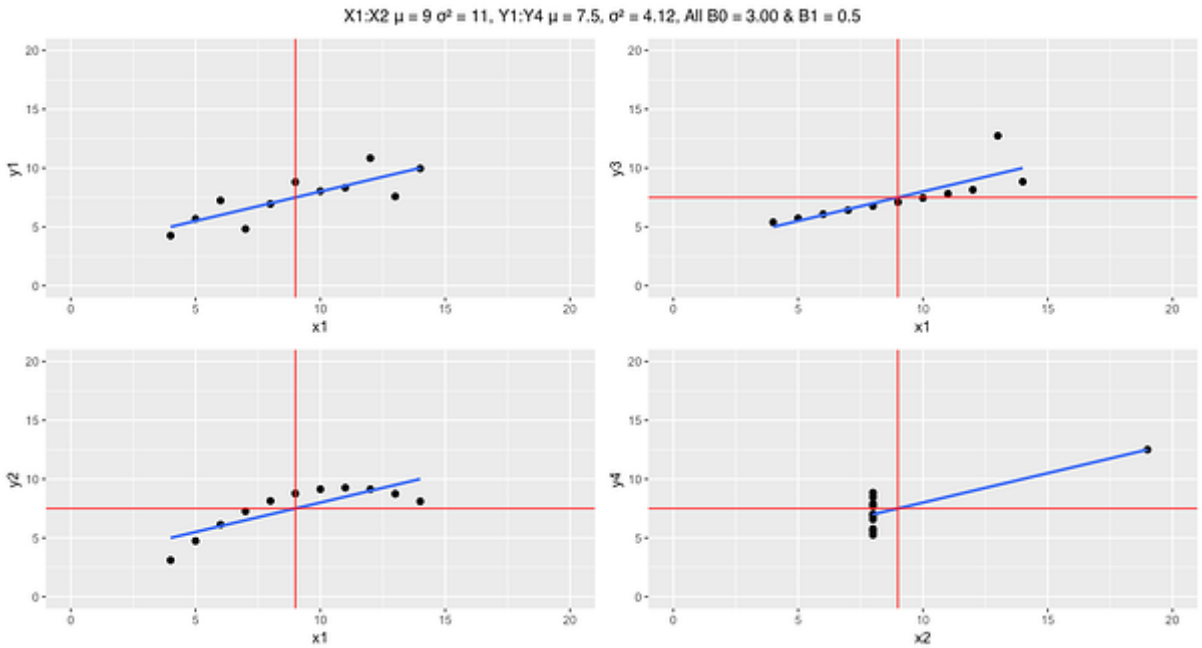

Figure 6: Anscomb's Quartet

Anscombe's quartet is a famous example of this, plotted above. However, many researchers have expanded on the principle in various ways (with datasaurus, a set looking like a t-rex being, the most humorous example).

7. Forgetting Statistical Fundamentals

The tremendous growth in easy access to data science tools is unquestionably a good thing for industry, academia, and the public sector. It's also unquestionably a good thing that access to the tools and knowledge have become democratized in a way that allows people from non-traditional pathways to become data science practitioners (as an economist, this is a self-serving assertion!). However, given the complexity of the subject matter and the broad array of academic backgrounds from which practitioners arise, it's easy to be spread too thin and begin to forget the principles underlying the tools being used.

Forgetting about the requirements inherent in, say, the Gauss Markov theorem, may distort your estimates by reducing efficiency or producing bias. Heteroskedasticity, for example, can distort a simple linear model, with the potential to be considerably more disruptive in nonlinear or time series models.

```
heteroskedasticity <- data.frame(
 x1 = seq(1000, 5000, length.out = 1000, sd=100),
 x2 = seq(1000, 5000, length.out = 1000, sd=100)
)
heteroskedasticity$x1 <- heteroskedasticity$x1 * rnorm(1000, mean=10)
p1 <- ggplot(heteroskedasticity, aes(x1, x2)) + geom_point() + geom_smooth(method='l

heteroskedasticity <- gather(heteroskedasticity)
p2 <- ggplot(heteroskedasticity, aes(value, color=key, fill=key, alpha=0.5)) + geom_

multiplot(p1, p2, cols=2, title = "X1's Variance Increases With its Level Value")
```

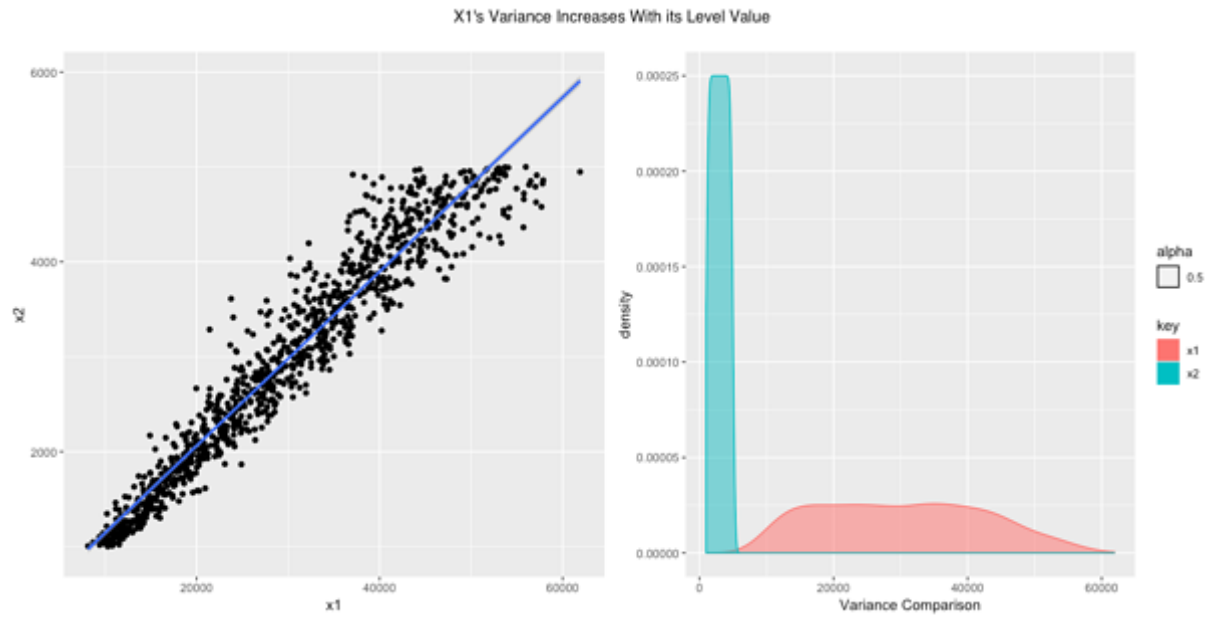

Figure 7: Artificially Generated Heteroskedasticity

8. Mathiness

One could view mathiness as the opposite of forgetting fundamentals, and it's arguably a bigger concern for team leads and end-product decision makers than it is for the individual data scientist. However, it is true enough that it's worth keeping in mind regardless of where you sit in the analytical supply chain.

Mathematical complexity does not mean something is better by default, and it can also be used - intentionally or unintentionally—to obscure what is really going on. My favorite illustration of this principle comes from John Siegfried's First Lesson in Econometrics, where he highlights different ways to represent 1 + 1 = 2.

$$1 + 1 = 2. \tag{1}$$

$$\ln e + (\sin^2 q + \cos^2 q) = \sum_{n=0}^{\infty} \frac{1}{2^n}. \tag{5}$$

$$e = \lim_{\delta \to \infty} \left(1 + \frac{1}{\delta}\right)^{\delta}, \tag{7}$$

$$\ln\left\{\lim_{\delta \to \infty} \left\{[(X')^{-1} - (X^{-1})'] + \frac{1}{\delta}\right\}\right\} + (\sin^2 q + \cos^2 q) = \sum_{n=0}^{\infty} \frac{\cosh p\sqrt{1 - \tanh^2 p}}{2^n}. \tag{12}$$

Figure 8: John Siegfried's First Lesson in Econometrics (Source)

As the adage goes: "*all models are wrong, some are useful*"... So it's important to remember complexity does not necessarily predict usefulness!

9. Overfitting

At its core, overfitting is a logic and reasoning problem because we need to appreciate that in every system there is some non-deterministic part of the system we don't and cannot necessarily understand (the universe is a complex place), which means mapping the past perfectly, almost by definition guarantees we're not understanding the future.

Figure 9: XKCD Curve Fitting (Source)

10. Making A Metric A Target

Economist Charles Goodhart, in which Goodhart's law is named after, noted that *"any observed statistical regularity will tend to collapse once pressure is placed upon it for control purposes."* Perhaps the most famous example of this is the (possibly apocryphal) British Cobra Effect. As the story goes, the colonial British wanted to diminish the problem of cobras in colonial India so they developed an incentive structure to encourage culling the population: paying locals to bring them dead cobras for a financial reward. Here, dead cobras are a metric tracking the rate of cobra culling, but by making it a target they distorted the incentives and encouraged more breeding of cobras. When they realized what this

management by numbers was doing, they scrapped the program resulting in all the cobra farms releasing their stock and a higher number of cobras existing in the wild than when they began the program (the story is of questionable historical accuracy, but there is a greater agreement that the French ran into a similar situation with rats in Vietnam).

Playing this out today, one can imagine a team of data scientists working to develop quantitative methods to optimize a reptile culling program using a suite of modern tools: time series decomposition to forecast cobra reproduction rates, building in seasonal trends and co-trending with grain harvests and field mice, using spatial density data to estimate the highest concentration of cobras to deploy field staff to pay out rewards, or developing behavioral models to tailor the financial reward amount by occupational concentration strengths within each region… and while they could follow the best statistical standards in the world, there wouldn't necessarily be any protection against the fatal flaw of having tied the program of reducing cobras to the tracking metric.

Less hypothetically, there are contemporary criticisms that this is a problem most modern economies are running into by focusing on GDP growth. What we as humans tend to care about are improvements in overall human well-being, and historically growth in GDP has been associated with overall improvement in human well-being, but some can credibly argue we've at least in part fallen victim to focusing on the metric instead of the end outcome.

Summary

There are similarities and differences between every data practice. Regardless of whether we're comparing the public and private sectors, finance and insurance, energy and transportation, or technology and healthcare industries, easy access to advanced tools has given us more power than ever to make informed, quantitative decisions. At the same time, it's also given everyone more options