Red Centre Software
Data Comes Alive

Automation: Paradiso, Purgatorio or Inferno?

Version: July 2017 (minor edits)

Dale Chant, Red Centre Software

John McConnell, Knowledge Navigators

ASC One Day Conference Series: Satisfaction Guaranteed? *The Challenges of Automation in Survey Research*

Thursday 11th May 2017, ORT House, 126 Albert Street, London, NW1 7NE

Abstract

We may all want to go to heaven (efficient, cost-effective, insight-driven, profitable, stress-free, happy staff) but too often end up in one of the other two places – a purgatory of endless struggle against intractable processes, or the hell of errors and systems failure leading to incorrect or useless analyses, missed deadlines, low morale and lost clients.

As software developers catering to high-end interactive and batched data analyses across a wide variety of locales and research cultures, we have witnessed all three outcomes as agencies are increasingly driven to automation by competitive pressures to deliver more for less.

This paper looks briefly at the trends in automation for the survey business over the last and present centuries, distinguishing micro-automation (eg using crosstab software instead of counting by hand), midi- (a processing pathway within applications on a single machine) and macro-automation (linked processes across disparate applications, multiple machines and even platforms, requiring human decision points), and proposes some basic principles and requirements based on practical experience for effective implementations.

For case studies, we look at how to automate: sentiment analysis of tweets using R (micro), personalised delivery of customised PDFs for each respondent comparing self against the rest (midi), and global tracking for standard reporting across N countries and regions (macro).

The primary principle is to automate sub-processes between (and never across) decision or diagnostic points, and the primary requirements are rich application APIs and sufficient in-house expertise to write and maintain controller scripts.

Introduction

The urge to automate is part of the human psyche, driven instinctively (I would argue) by the needs of self-preservation. An inefficient tribe hunts one-man, one prey. A more efficient tribe hunts big prey by coordinated action. A superior tribe automates hunting by driving prey *en masse* over a cliff, potentially supporting a greater population via food and materials, and freeing labour resources for other areas of advancement, such as the acquisition of knowledge, skills and technology for defense against enemies, territorial expansion or cultural advancement.

With enemies at bay, and territory acquired, attention can turn to internal efficiencies. Roman aqueducts used gravity to support whole cities, whereas human (usually female) water-carriers can support a village at most. Wind replaced rowing for water transport. Dutch windmills to this day pump groundwater to increase viable land, and millers invented water wheels to grind grain. Exploitation of gravity and wind may be free, but human effort by slavery or a huge peasant class remained the main way of getting things done until the industrial revolution. From electricity alone, an average middle class western individual now utilises the energy equivalent of 100 full-time personal slaves per day [1].

Without venturing into the theoretical swamplands, the key point about automation is always to reduce or eliminate human effort, for at least the same or more/better outputs. An industrial example would be robotic automobile manufacture – what previously required a cast of thousands of assembly line workers delivering varying quality, now needs only a few to monitor and maintain the robots, for near absolute identical and better quality unit outputs. An every-day example is boiling the kettle – using electricity to automate the task, thereby eliminating the effort of chopping the wood, lighting the fire and cleaning up.

Automation in Market Research

For market research, the primordial instincts are manifest in the need for business survival. Like sharks who stop swimming, stasis is the kiss of death by competitive pressure, and so agencies are driven to innovate for efficiency or fall eventually by the wayside. There are no agencies still tabulating from Hollerith cards.

The two primary candidates for innovation with regard to automation are data collection and data processing.

Within most of our professional lives, data collection has moved predominantly from clip-board intercept to self-administered by internet or robo-callers. This has certainly satisfied the requirement of reduced human effort for more output (completed questionnaires), but there remain serious questions about the impact on data quality [2].

On the data processing front, however, there is no doubt that far more is now obtained by far less, and the ways in which this has been achieved, and the considerations for when, where and how to implement automation, are the main focus of this paper.

Advantages, Disadvantages, Risks

Advantages:

1. Reduced human effort
2. Reduced labour costs
3. Reduced running expenses
4. Reduced time to task completion
5. Repeatability
6. Scalability
7. Increased productivity
8. Increased efficiency
9. Increased quality
10. Enforced consistency
11. Ability to perform tasks previously impossible
12. Increased opportunities for staff redeployment

Items 1 to 6 lead to items 7 and 8, productivity and efficiency – more for less. Items 9 and 10, quality and consistency, lead to better for less. Items 11 and 12 can impinge on strategic thinking and business decisions. Good examples for the previously practically impossible include data mining by multi-dimensional analysis, correlation of all variables against each other, text-searching millions of records, and development of norms across many surveys. For simply impossible by human effort alone, we now have things like automated dynamic reporting to on-line portals, turn-around time for DP to analysis reduced from days to minutes, even seconds, real time aggregations and panel management and scrutiny.

Disadvantages:

1. Reduced flexibility to meet ad hoc requirements
2. A broken process can run amok, wreaking havoc
3. Skilled staff required to write system-wide and application-specific scripts
4. Skilled staff required to monitor operations and maintain scripts
5. Expensive to implement

Reduced flexibility can be controlled to some extent by design, but the more flexible (in terms of types of inputs and outputs), the more complex, increasing development time and the potential for end-user operational errors. The point is to resolve many small (manual) tasks to a single automated one, so the opportunity to do a small task differently to meet an ad hoc requirement is necessarily obviated.

Unlike humans, an automated process is blind to consequences, and will endeavour to run to completion regardless. If an error state is not suitably trapped, a script will plough on, and could write garbage files, or corrupt good ones, or, worse, introduce subtle errors no one notices except the client's CEO at your showcase presentation.

Items 2, 3 and 4 encapsulate the so-called Paradox of Automation:

> *[T]he more efficient the automated system, the more crucial the human contribution of the operators. Humans are less involved, but their involvement becomes more critical. If an automated system has an error, it will multiply that error until it's fixed or shut down. This is where human operators come in. Efficient automation makes humans more important, not less.* [3]

Items 3 to 5 are considerations for management. Depending on the automation project, the costs to employ, train and implement to the requisite levels may obviate the anticipated advantages.

Risks:

1. Catastrophic failure
2. Reruns due to machine or human errors obviate time savings
3. Insufficient checks for correctness
4. More work is invented to take the place of automated tasks
5. The techno-phile urge to tinker
6. Staff mobility can create unexpected skill deficits (usually just when things go awry)
7. Over-confidence in success leads to devaluing the need to understand the processes
8. Cost to implement and deploy exceeds benefits
9. Failure to accept diminishing returns
10. Over-automating
11. Over-ambitious goals (eg AI and natural language processing)

Unfortunately, catastrophic failure can happen any time, quite out of the blue. Automation puts many eggs into one basket, and if the basket breaks, so too do all the eggs. The only safeguard against this is sufficiently skilled personnel on hand to diagnose and fix, if possible. A major cause of catastrophe is internet outage on systems which collect/collate external data inputs, in which case all you can do is wait.

Reruns due to machine or human error are usually learning experiences, and should diminish with time and experience. Wherever possible, novel processing errors should be trapped for future runs by extending the checks for correctness.

Items 4 and 5 are personnel issues. If the automation is successful, then resource utilisation gaps will appear. How to fill the gaps should be considered by management – if left to their own devices, academically inclined analyst/researchers and tech-savvy DP will tend to futz and fiddle at the expense of the major goals. This is not necessarily a bad thing, but given the inevitability of Parkinson's Law (work expands to fill the time available [4]), pre-emptive planning is much advised. The techno-philes are often the ones who implemented an automated system, but once working to standard, their creative impulses are better directed at a new project, where previous lessons can be applied, rather than pursuing ever smaller incremental improvements (see also

item 9). Improvements mean changes, and changes are risky and can unsettle users. *If it ain't broke, don't fix it*.

Items 6 to 11 are more in the management domain.

Item 6 is a common scenario: After some months to bed down a system, requiring training on the agency side, suddenly key staff have moved on or been redeployed, and their replacements haven't a clue. If the system is fixed inputs->black box-> fixed outputs, the inputs are contracted to conform, and no software issues, then anyone can push the button and collect the outputs, but these pre-conditions cannot be assumed or guaranteed. Your field supplier may be having staffing or technical problems, and a data wave is delivered which breaks all the agreed rules. The executing machine may have been subjected to an overnight stealth update which breaks the black box. The black box may fail because the designer did not anticipate unlikely but legal inputs. And plain bugs are present in all software beyond the trivial, whether known or not. When delivery pressures collide with failures in an automated system, staff who understand the system well enough to fix it or to devise impromptu work-arounds are de rigueur, or business damage will ensue. Note that by 'staff', we mean the plural. Redundancy is essential, to cover availability, absenteeism, leave, resignations, etc.

Overconfidence can lead to replacing the experts with juniors, usually a false economy.

Cost/benefit depends on accurate appraisal of the task.

Diminishing returns is to be expected as the automation agenda reaches the currently prevailing technological limitations. This phenomenon is described by the logistics curve [5]. Management, heady with success, demands more of the same, but there is no further scope for realistic improvements, resulting in more and more effort for less and less benefit.

Over-automating is trying to go a bridge too far. This risk can manifest in trying to use expert systems to eliminate human decisions. Macro-automation systems, like a global tracker, require many decision points (such as how to implement and process questionnaire changes), and it is not feasible to create a comprehensive set of rules to cover all contingencies. To be useful, tracking must be dynamic and adaptive, hence, the rules on how to implement change or fix mistakes cannot be predicted.

Artificial intelligence (AI), machine learning, and natural language processing (NLP) are still highly experimental, so much care is advised. Good classification of verbatims depends on unrealistic training samples. 100,000 is common; but there is no point in manually coding 100,000 training verbatims to be applied on a survey of 10,000 respondents, and where/how do you obtain the training data? Sentiment scores and similar from social media text mining are full of traps [6]. Neural net decisions cannot be explained nor justified [7].

Overall, there are far more risks than disadvantages (Inferno), and the risks can be mitigated by appropriate management (Purgatorio, strive to purge the glitches), so the overall case for automation – it can get you to Paradiso – is very good, and empirical observation of market research agencies leaves no doubt that the general consensus is that automation should be deployed wherever it is feasible and practicable to do so, whatever the pain may be along the way.

A Brief History:

Manual Tabulation

The earliest formal surveys I am aware of come from Athenian democracy. The simplest survey was a yes/no on assembly resolutions, recorded by pebbles (psephai), black=no, white=yes.

Q1. Despite our ruinous war with the Spartans, do you agree that we should invade Sicily?

Yes (white): 85%

No (black): 15%

This yields a dichotomous variable, with data processing as sort and count.

A more complicated codeframe arises from the procedure for ostracism. Once a year, the citizens could nominate an individual for ten-year exile, highest vote is shown the door.

Q2. Please scratch the name of your choice for this year's ostracism on the supplied pottery fragment (open-ended, you may cite reasons)

Themistokles: 18,345

George: 78

Peter: 35

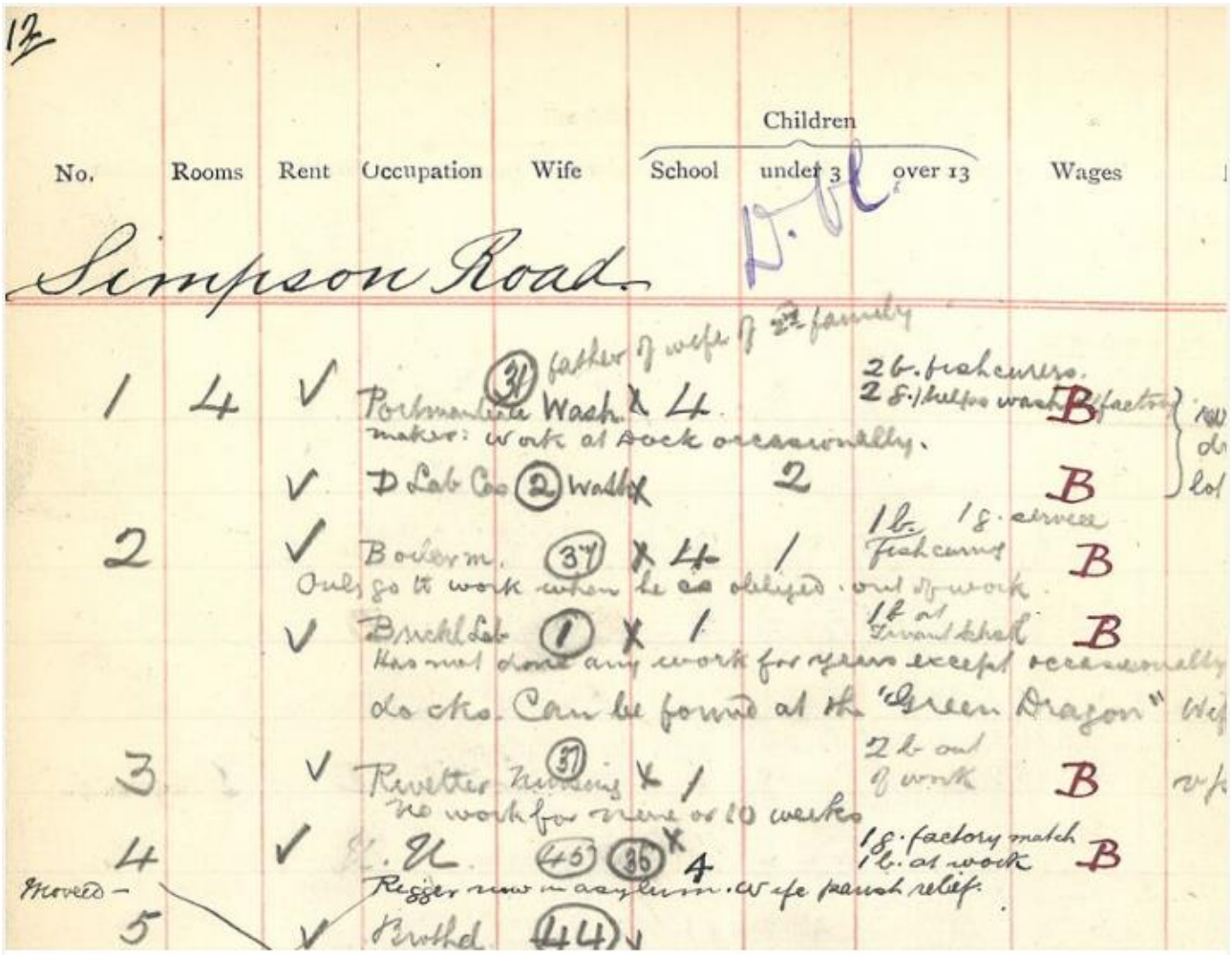
Basil: 10

…citizens gave the name of those they wished to be ostracised to a scribe, as many of them were illiterate, and they then scratched the name on pottery shards [ostraka], and deposited them in urns. The presiding officials counted the ostraka submitted and sorted the names into separate piles. The person whose pile contained the most ostraka would be banished. [8]

This describes a sorted histogram of a categorised verbatim variable.

Charles Booth

Moving on about 2,300 years to late 19C England, it was Charles Booth (president of the Royal Statistics Society 1892-84) who first conducted what we would recognise as a modern social survey, *Life and Labour of the People in London* [9]. This was a massive study. How did Booth get from the questionnaires [10]:



A notebook with data about the residences of Simpson Road

To final tables?

TABLE III.—*Earnings (for one week) in various Employments, compared with conditions as to crowding.*

TRADE SECTIONS.	NUMBERS.		PROPORTIONS.		
	Adult Males employed.	Scheduled.	Wages under 25*s.*	Crowded (employees only).	Wages under 30*s.*
			per cent.	per cent.	per cent.
Building trades	*97,873	5,066	10½	45	40
Cabinet makers, &c.	29,515	591	14	52	30
Carriage building	7,348	685	34	41	51
Coopers	2,978	367	8	38	28½
Shipwrights, &c.	1,813	140	18½	24	25
Sundry workers in iron and steel	36,702	13,203	32	36	46
Brass, copper, tin, lead, &c.	11,130	1,402	24½	49½	43½
Jewellers, &c.	4,748	412	3½	28	16½
Watches and clocks	2,143	147	20	28	36
Surgical, &c., instruments	5,184	830	10	30	25
Musical instruments and toys	5,885	208	17	37	30

TABLE IV.—*Showing for each section the proportion of heads of families born in London, as compared to those living in the Inner Ring or under crowded conditions.*

SECTION.	Born in London	Living in Inner Circle.	Crowded.	SECTION.	Born in London.	Living in Inner Circle.	Crowded.
	Per cent.	Per cent.	Per cent.		Per cent.	Per cent.	Per cent.
Bookbinders	81	58	41	General labourers	52	37	58½
Paper manufactures	78	60	49	Dock and wharf service	52	42	25
Brushmakers	76	57	40	Extra service	51½	43	40½
Lightermen	75	52	35	Publicans	51	47	10
Glass & earthenware	71	52	47	Commercial clerks	51	19	10½
Musical instruments and toys	71	30	32	Grocers	50	34	15½
Stationers	70	35½	17	Engive drivers, &c. (undef.)	49	36	37
Coopers	69	54	36	Civil and municipal			
Trimmings, &c.	69	56	36				
General shopkeepers	68	58	40				

This is near 100,000 respondents, and the total sample was 130,000 households – surely impossible as hand tabs? All Booth has to say on the matter is

> *From such notes I, with the assistance of my secretaries, **tabulated the information given in our schedules**, each of which represents an immense amount of labour in collating: and from them, also, the map was made which fronts the title-page. The people - those of them with school children, concerning whom only we had information - were classified by their employment and by their apparent status as to means; the streets were classified according to their inhabitants. Such is the nature of our information, and such the use made of it.* [11]

After quite some research (automated from my desk by internet search, and quite impossible pre-late 1990s), it would appear that manual tabulation was indeed the order of the day. Kevin Bales' PhD thesis was on Booth. He says

> *Imagine the hand tabulation and aggregation of data on some 130,000 households! While the provenance and interpretation of the poverty data has been debated, no one has found errors in the statistics themselves, though the lack of multivariate techniques plagued Booth.* [12]

This requires faculties steeled in mental arithmetic and attention to detail beyond our modern ken. In a modern context, this is a survey which would never happen but for automation – there is not the personnel or funding to ever consider manual data processing to final tables. Structurally, this survey is a levels job of Households/Members - the most intractable of forms for both data processing and analysis.

Early Automation

The earliest appearance of automation in survey data processing I can find is the original Hollerith counter used for the 1890 USA census.

This equipment is representative of the tabulating system invented and built for the US. Census Bureau by Herman Hollerith (1860—1929). After observing a train conductor punching railroad tickets to identify passengers, Hollerith conceived and developed the idea of using punched holes to record facts about people. These machines were first used in compiling the 1890 Census. Hollerith's patents were later acquired by the Computing-Tabulating-Recording Co. (which was renamed IBM in 1924) and this work became the basis of the IBM Punched Card System. Hollerith's tabulator used simple clock-like counting devices. "Then an electrical circuit was closed (through a punched hole in a predetermined position on the card), each counter was actuated by an electromagnet. The unit's pointer (clock hand) moved one step each time the magnet was energized. The circuits to the electromagnets were closed by means of a hand—operated press type card reader. The

operator placed each card in the reader, pulled down the lever and removed the card after each punched hole was counted. [13]

Following Booth, and into the 20C, surveys were increasingly acknowledged as vital to good governance and effective business decisions, but few practitioners availed themselves of automatic counters. From Jean Converse:

> *Tabulation itself was something of moment. Keypunch machines and counter sorters were available to the larger firms of this period, but it was not until the late 1930s that verifying machines were in any considerable use, so errors were difficult to detect. Hand tabulation was more practical in many situations, especially when there were few cross-tabulations.* [14]

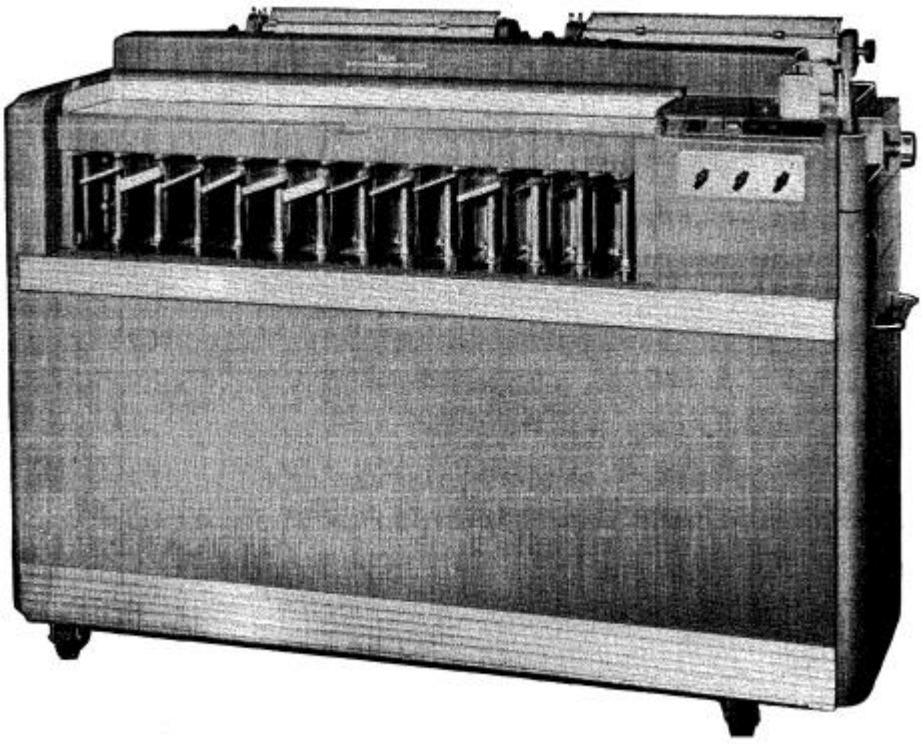
> *This published research was largely the fruit of the precomputer era. Data analysis was crafted with paper and pencil, desk calculators, the counter sorter of Hollerith cards, and the IBM 101, which could be programmed by wiring a simple board to carry out various cross-tabulations. This technology was the most advanced equipment in the "machine rooms" of all three major organizations during most of these early years. Correlation, regression, analysis of variance, and factor analysis were only rarely carried out on the desk calculator.* [15]

The IBM 101 was state of the art pre-computers. For the US 1950 census, the following machines were employed, including 45 101s [16]:

THE CENSUS OPERATION

Table E.--Machines Used for Punching and Tabulating Operations, 1950 Censuses

Machine type	Maximum number	Census of Population	Census of Housing	Census of Agriculture
IBM Numerical Punch #016.....	1,445	1,015	459	46
IBM Numerical Punch #024.....	691	167	20	530
IBM Alphabetical Duplicating Punch #031	11	8	3	-
IBM Punch Card Verifiers #055.....	159	6	2	151
Census Verifiers #280.....	709	364	210	135
Census Unit Count Machine #581.....	32	27	24	-
Census Multi-Column Sorter #488.....	28	27	15	2
Census Gang Punch.....	12	8	4	-
Census Recode Machine.....	1	1	-	-
IBM Duplicating Summary Punch #524.....	84	30	30	34
IBM Sorter #082.....	103	23	45	36
IBM Electronic Statistical Machine #101	45	28	17	19
IBM Alphabetic Accounting Machine #402.	60	12	24	35
IBM Accounting Machine #407.....	5	1	-	5
IBM Electronic Calculating Punch #604..	6	1	5	1
IBM Reproducing Punch #514.....	82	20	40	34
IBM Collator #077.....	33	4	16	13
IBM Alphabetical Interpreter #552.....	1	1	-	-
Richards Copyholder.....	900	900	-	-
Pres-to-line Copyholder.....	900	491	535	-
Agriculture Copyholder.....	785	-	-	785



IBM 101 Electronic Statistical Machine

From the official 1958 manual [17]:

The 101 Electronic Statistical Machine combines in one unit the functions of sorting, counting, accumulating, balancing, editing, and printing of summaries of facts recorded in IBM cards.

> *The following operations may be performed at the rate of 450 cards a minute:*
>
> *1. Sort IBM cards in numerical or alphabetic sequence.*
> *2. Arrange cards into any desired pattern.*
> *3. Check cards for consistency of coded information.*
> *4. Check the accuracy of sorting.*
> *5. Search files of cards for specific facts or combination of facts.*
> *6. Count cards for as many as 60 different classifications in one run.*
> *7. Add two 5-digit amounts punched in IBM cards to accumulate two 8-position totals; or add one 9-digit amount to accumulate one l2-position total.*
> *8. Print results in final form on one or two reports of convenient size.*
> *9. Print group identifications.*
> *10. Print a check symbol on each line of the report to indicate that the totals printed on the line cross-check.*
> *11. Summary punch totals in IBM cards when one or two summary punches are connected to the 101.*

Sounds good. That's a lot of otherwise manual tasks now being automated. Apart from being frequencies only, this is a modern-looking machine-generated table:

General Mfg. Corp. Endicott, N. Y. Part I

Left Carriage

Report — Employees By Plant and Age Group — Male

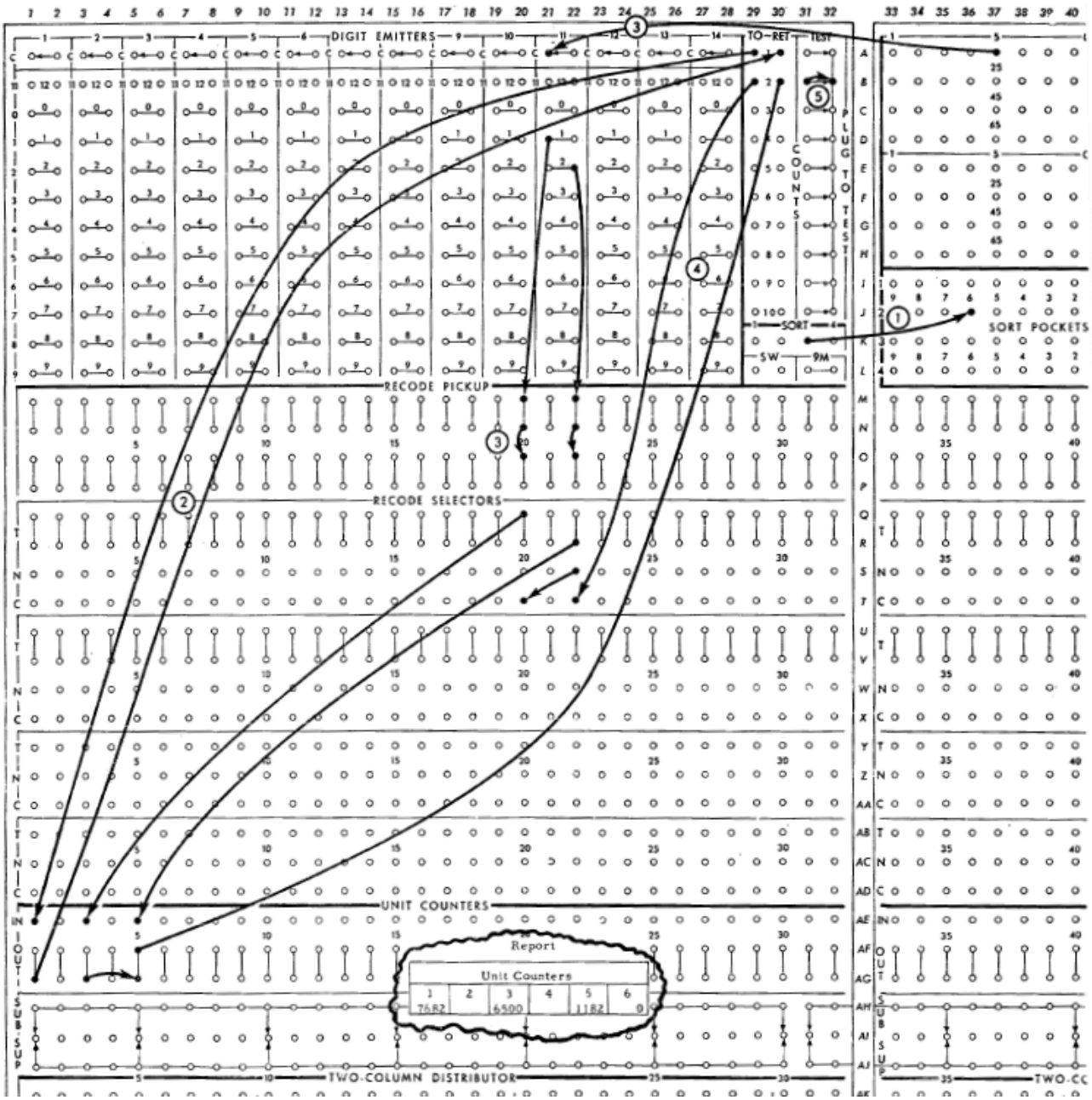
Total Earnings — Male	Plant No.	Total Male and Female	Total Male	Male Age Groups							0 Check
				15-19	20-24	25-29	30-39	40-49	50-59	60-64	65 And Over	
		1	2	3	4	5	6	7	8	9	10	11
44511497		8359	6182	160	875	902	1184	1881	895	283	2	0
← Accm. Counter →	Group Indic.	← Unit Counters →										
21811143	1	4096	3029	78	429	442	580	922	438	139	1	0
10682519	2	2006	1484	38	210	216	286	451	215	68		0
9792301	3	1839	1360	35	192	199	261	414	197	62		0
2225534	4	418	309	9	44	45	57	94	45	14	1	0

[18]

But the reality was rather messy. In summary, the patch bay steps to specify the above table are

> *COUNTS To hub 1 would be wired to cause counting in both counters 1 and 31;*
> *COUNTS To hub 2 would be wired through recode selectors so that it would count males in counter 2 or females in counter 32;*
> *COUNTS TO hub 3 would be wired first through age-group distributors and then through male-female recode selectors to count males in one of the counters 3-10, or females in one of the counters 33-40.*

To obtain just counts for Gender, the full patch is [19]



One can see the problems with this approach. Better than hand tabulation? Most certainly, but there is still a long way to go.

Modern Automation

The game-changer, of course, was digital computing and software. Converse:

> *Giant computers, which began to be installed in major universities in the late 1940s and early 1950s, did not accommodate analytic routines for social science data until the late 1950s and early 1960s. A few intrepid social scientists, well counseled by their statisticians, set upon the giant central computers (such as the IBM 650) to wrest from them a few key analyses, such as some multiple regressions, but computer technology was still forbidding and inhospitable for social science use. The books and articles these researchers published around 1960 were, indeed, the last of the handcrafted work.* [20]

> *Three new features were to come in the future. First, there would be machine technology: the computer models that would revolutionize the kinds of analysis that could be performed on survey data, as well as the speed. Prodigious new programs would be devised that could "ransack" data for relationships and test elaborate causal models.* [21]

But there was a downside.

> *Computer technology, for all its power, had the major defect in the 1960s and 1970s of intruding between the analyst and the data. Analysts delivered their "batch" to the queue at a computing center, and if all went well they got their output a few hours or a day later. This put a crimp in the style of those artists in data analysis who liked to work with data as Stouffer did, turning at once to the counter-sorter to try out an idea or resolve an argument, intent upon the immediate detective work of survey analysis.* [22]

But by the 1970s, into 1980s, with the advent of remote terminals and micro-computers, the spontaneous pursuit of hypotheses again became feasible, and a slew of software packages appeared which could automate the generation of vast numbers of tables. Some I am personally familiar with or have used operationally include: Surveycraft, Quantum, UNCLE, CfMC, Merlin, Bellview. But these systems all came with their own proprietary specification language, often requiring some years to achieve deep expertise.

By the 1990s and the advent of GUI operating systems , desktop cross-tabulators appeared which enriched the automated outputs (usually managed by a dedicated DP group) by further allowing interactivity (Quanvert, InfoTools, MarketMind, MI-Pro, Asteroid, mTab,…). For the first time, non-technical analysts and researchers could safely effect at least some traditional DP tasks, such as rerunning a set of tables with different banners, under arbitrary demographic splits, with different weights applied, etc.

Concurrently, advanced statistical packages like SPSS, MatLab and SAS all supported at least rudimentary cross tabulation, and could accept any importable matrix as an input to automated statistical analysis.

In the current century, and particularly the last fifteen years, desktop crosstab specification languages all build on the MS COM automation platform, with VBScript/VB.Net (Dimensions, Ruby, modern SPSS among others) or JScript (Askia, Q, Blaise, among others) as the underlying execution engine. The respective APIs are necessarily quite distinct, but sharing the engine with the Windows operating system facilitates considerable synergies: VB and JScript syntaxes are common to all implementations, and automating the tabulator/analysis software in concert with MS Office for client reporting from a single script becomes trivial. The pool of potential scripters has been effectively widened to anyone who can write an Excel macro. Understanding SPSS syntax is no help whatsoever for learning how to write Quantum/Merlin/UNCLE/CfMC specs, but knowing how to write an MS Office macro is most of the battle already won for Dimensions or Ruby scripts. We now routinely observe analyst/researchers performing tasks which last century would have remained the sole preserve of DP.

Micro-Automation

A micro-automation is a single process which runs on a single machine to completion with no interruption. We all use these countless times a day in the course of ordinary life, both personal

and professional: change font for an entire document, search/replace in a document, search email, internet queries, recalculating a spreadsheet, charting a matrix, OCR (used to capture all quotes in this document), etc.

Cross Tabulation

The relevant case in point is a cross tabulation. Hand tabulation or hole counting is now completely historical – all case data is machine readable, and all tabulation is done by software.

Compared to Booth's day, automated cross tabulation has proved the greatest reducer of every-day labour in survey data processing and analysis, and within that limited scope, Paradiso has been achieved – all the advantages are fulfilled.

The only real risk here is Parkinson's Law – interactive desktop cross tabulation is now so easy that we tend to generate a great deal more of them than would ever have been possible using the IBM 101, too often without much regard to informational content. Clients bear at least some responsibility for this. Often, when we ask "tell me again why you need 653,421 hard copy tables per month", the response is "the client insists". When we ask "and what happens to them?" We are told "they sit in boxes in a warehouse". Excel for KPIs instead of hard copy is now common, but that makes the problem worse – one can no longer plead the physical difficulties of printing, but even on a 16 gig 64 bit machine there is a limit to how many ways you can split a hundred measures by brand within demographics. A thousand sheets at a hundred thousand rows each is asking for trouble.

Twitter Sentiment Report

So for a case study, we look at something more interesting from the bleeding edge: searching Twitter for key terms, with outputs as the retrieved cases (as a text file), a word map, a histogram of emotional valency, and a bar chart of sentiment (as a PDF doc). This sounds ambitious, but the marvelous R makes it trivial, the full script (omitting my confidential access tokens and login authorisation, requiring 9 lines) being less than a page:

```
searchterm <- "#trump"  #search term, use + to separate terms
num        <- 1500      #number of tweets to return

#Search tweets
list <- searchTwitter(searchterm, n=num, lang="en", since=NULL, until=NULL,
                      resultType = "recent", retryOnRateLimit=150)

#save raw tweets to file
sink(paste("c:\\RScripts\\Sentiment\\rawtweets.txt"), split=TRUE)
list
sink()

library("wordcloud")                              #wordcloud
library("tm")

l <- sapply(list, function(x) x$getText())        #clean up list
l <- iconv(l, "latin1", "ASCII//TRANSLIT")
l <- iconv(l, to='ASCII//TRANSLIT')
lc <- Corpus(VectorSource(l))                     #create corpus
lc <- tm_map(lc, content_transformer(tolower))    #Convert every word to lower
lc <- tm_map(lc, removePunctuation)               #Remove punctuation
lc <- tm_map(lc, function(x)removeWords(x,stopwords()))  #Remove stop words

library(RColorBrewer)                             #set palette
pal2 <- brewer.pal(8,"Dark2")
```

```
pdf(paste("c:\\RScripts\\Sentiment\\tweets.pdf")) #all outputs to a PDF
wordcloud(lc,min.freq=num/50,max.words=500, random.order=T, colors=pal2)

library(syuzhet)                                    #sentiment library
s_v <- get_sentences(l)

#plot Emotional Valence as bars and scatter
s_v_sentiment <- get_sentiment(s_v, method="syuzhet")
plot(s_v_sentiment, type="l", main="Emotional Valence per Tweet",
     xlab = "Narrative Time", ylab= "Emotional Valence")
smoothScatter(s_v_sentiment)

#nrc sentiment analysis
nrc_sentiment <- get_nrc_sentiment(s_v)
barplot(sort(colSums(prop.table(nrc_sentiment[, 1:10])*)),
        horiz = TRUE, cex.names = 0.7, las = 1,
        main = "NRC Sentiment", xlab="Percentage")

dev.off()  #close PDF
```

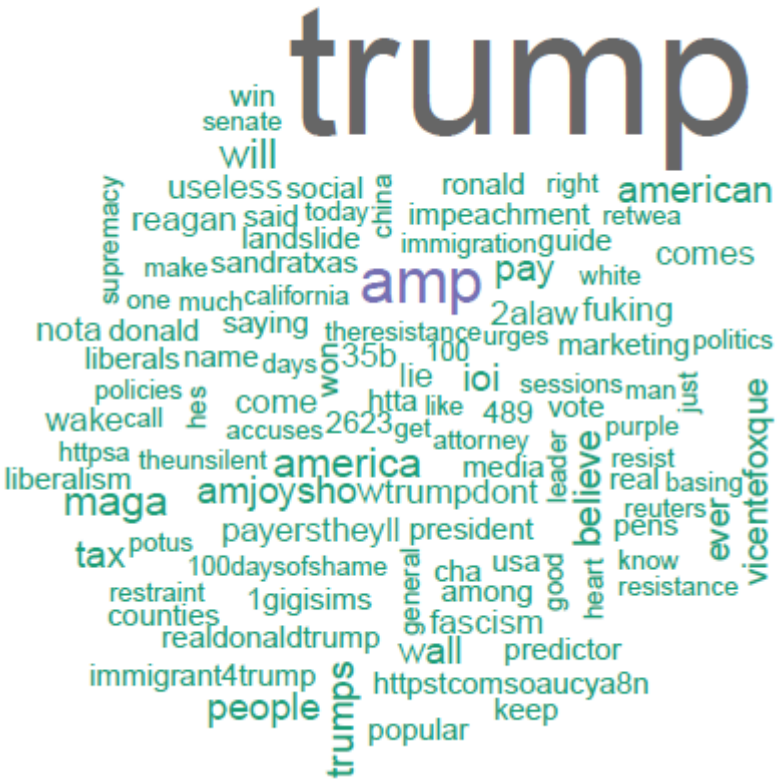
The above was patched together from the various examples [23], and took about an hour to get going. You do not have to be an R expert – I certainly am not – just basic familiarity with R is usually sufficient.

The outputs are

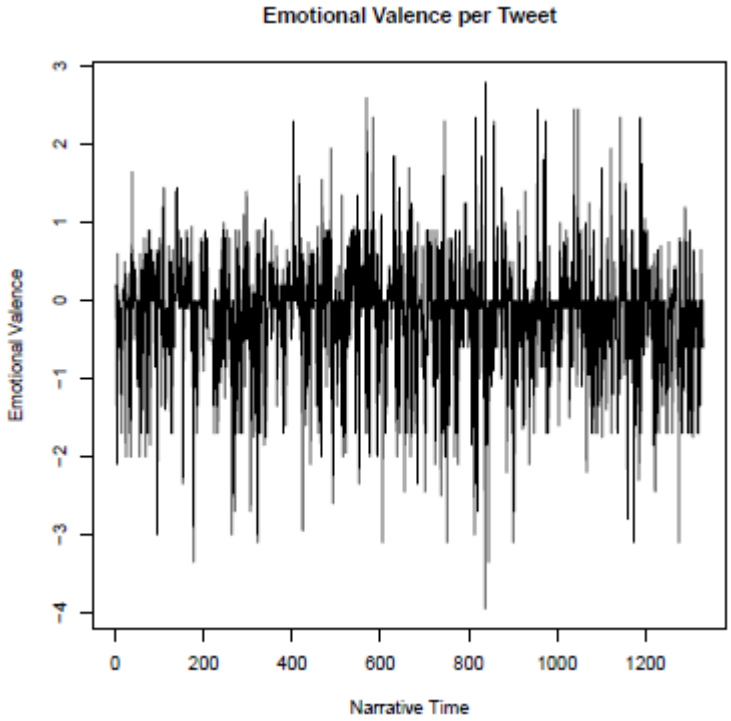
Text dump:

```
[[1]][1] "ProSyn: #Trump's Afghanistan bombing was an example of letting m
[[2]][1] "news_political_: Uber's Kalanick got caught for tricking Apple,
[[3]][1] "AlanDKirby: RT @2ALAW: Ronald Reagan:\n\"If Fascism Ever Comes I
[[4]][1] "hookem5746: On #Election Day it was D +3 . New #WaPo #poll (D+7)
```

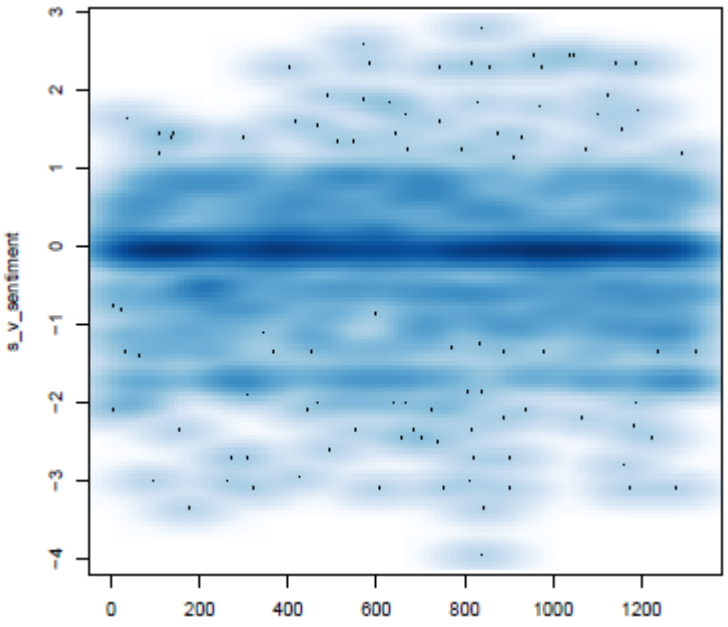
Word map:



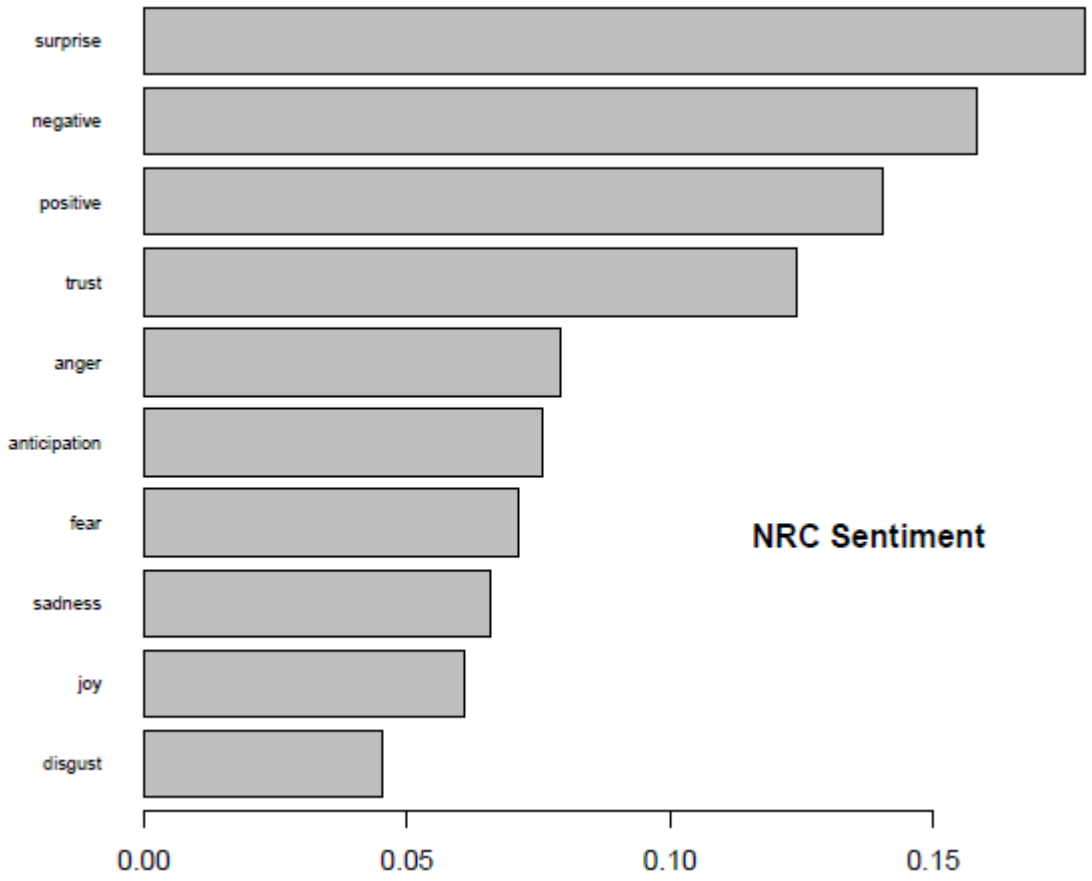
Valency:



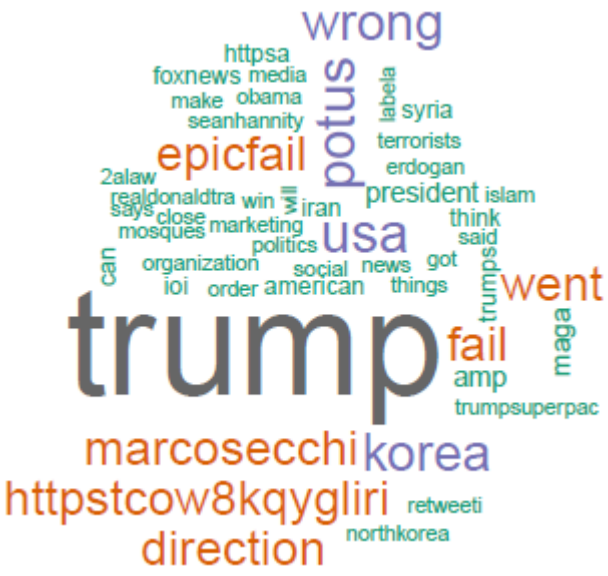
As smoothed scatter plot:



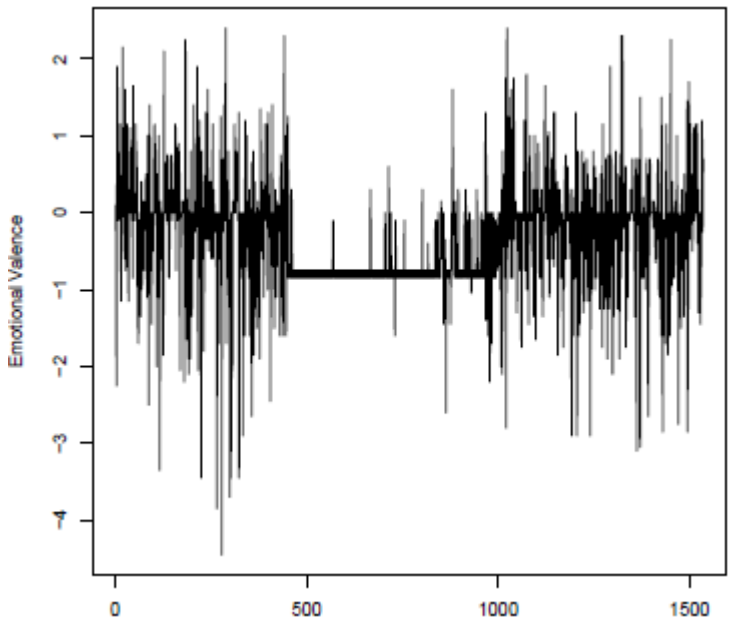
NRC Sentiment:



This is a lot of deliverable for very little effort, but care is required in interpretation. Twitter is biased left because the left agitates while the right stays home being comfortable, so the NRC negative exceeding positive is not surprising, but there is no way to tell who is the target of the negativity. The only true conclusion is more negative sentiment than positive, but that can itself be skewed dramatically by tweet bots and automated retweets. A good example is



This looks good – we have some key themes and issues, but the set of input tweets has been bot-corrupted, as the Valency chart shows:



This is not indicative of a human response pattern, and explains why *epicfail* is prominent in the map - the original tweet being retweeted several hundred times:

```
[334] "RT @marcosecchi: They went the wrong direction ! #usa #epicfail #fail #trump #korea
[335] "RT @marcosecchi: They went the wrong direction ! #usa #epicfail #fail #trump #korea
[336] "RT @marcosecchi: They went the wrong direction ! #usa #epicfail #fail #trump #korea
[337] "RT @TrumpSuperPAC: Facebook Promotes Murder, Gives Employees Permission to Skip Work
[338] "RT @marcosecchi: They went the wrong direction ! #usa #epicfail #fail #trump #korea
[339] "RT @marcosecchi: They went the wrong direction ! #usa #epicfail #fail #trump #korea
[340] "RT @marcosecchi: They went the wrong direction ! #usa #epicfail #fail #trump #korea
[341] "RT @marcosecchi: They went the wrong direction ! #usa #epicfail #fail #trump #korea
[342] "RT @marcosecchi: They went the wrong direction ! #usa #epicfail #fail #trump #korea
[343] "RT @marcosecchi: They went the wrong direction ! #usa #epicfail #fail #trump #korea
[344] "RT @marcosecchi: They went the wrong direction ! #usa #epicfail #fail #trump #korea
[345] "RT @marcosecchi: They went the wrong direction ! #usa #epicfail #fail #trump #korea
[346] "RT @marcosecchi: They went the wrong direction ! #usa #epicfail #fail #trump #korea
```

There is no point whatsoever in auto-data-mining data which was itself auto-data-generated. For more on tweetbots and auto-data generation, see Chant [24]

The Twitter search and chart automation has all the advantages, at the risk of over-ambitious goals.

Midi-Automation

Midi-automation is a batch of micro-automations, not necessarily of the same type, which can also run to completion without interruption. A simple example is generating a set of tables and exporting them to Excel, coordinated by a script. Time from inception to completion is linear with respect to the number of tables. No in-process decisions are required.

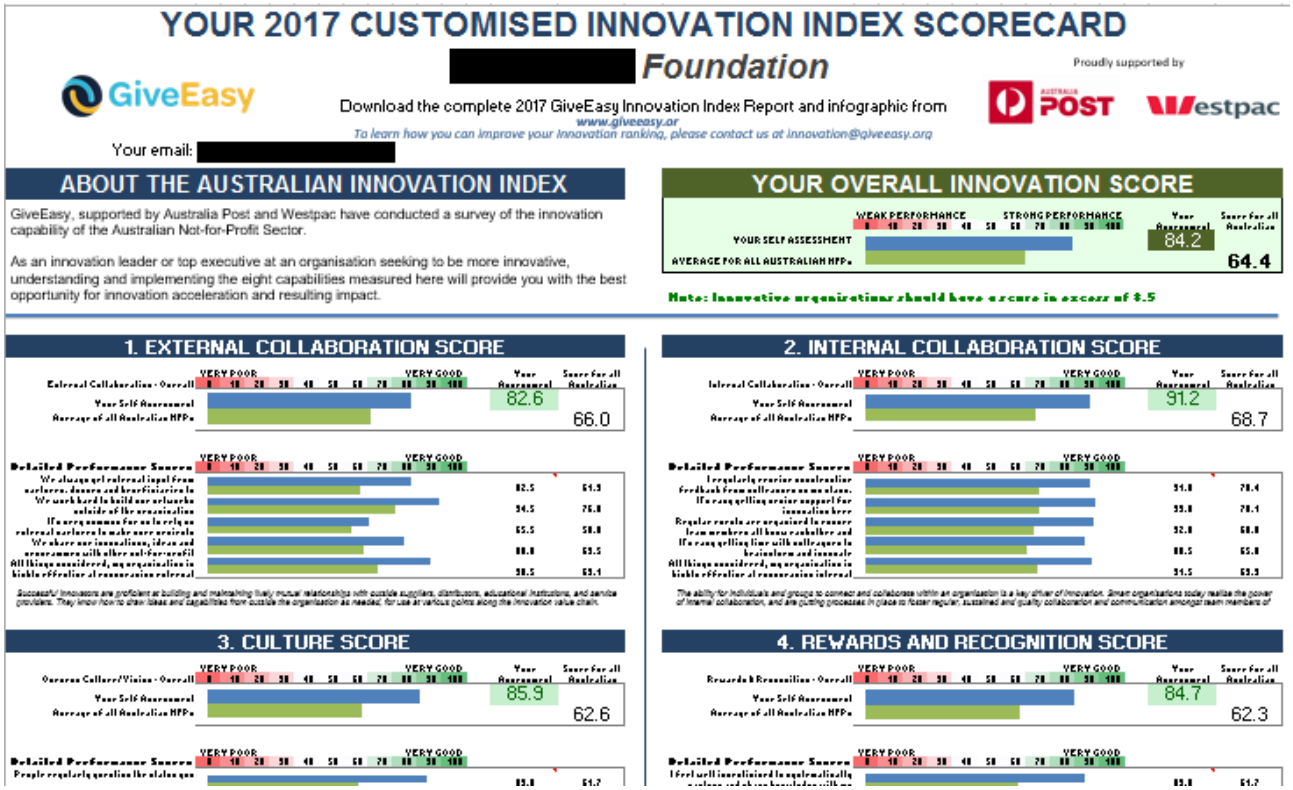
Personalised Document Delivery

Here is a more complex example, as a case study.

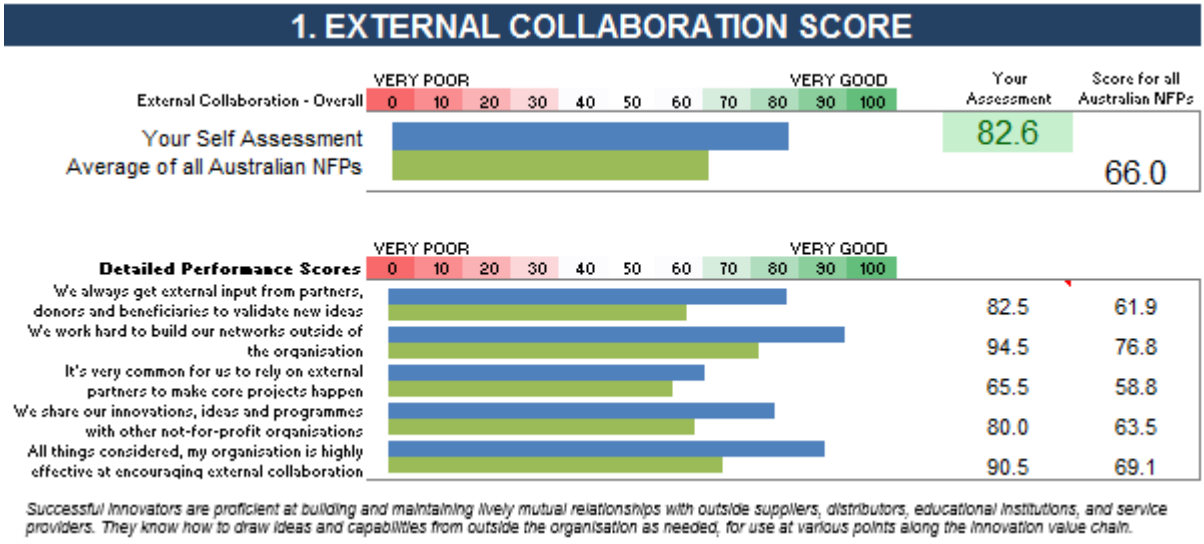
The job is a survey of not-for-profit charities, looking for factors which drive innovation in fund-raising and service delivery. The respondent incentive is a personalised report, showing the respondent's organisation's scores against the averages of all other organisations. Some organisations have only a single respondent, most have a few, and some have many. Total sample

is around N=1,500. The deliverables are the individualised report as a two-page PDF attachment, formatted in and saved from Excel, to each respondent's email.

The master XLSX (top half of page 1) is



There are eight score panels, each of the form



Each panel comprises five related factors, which are single-response rating variables, scaled up from 0 to 10 as 0 to 100.

1500 respondents * 8 panels * 5 statements * 2 columns (self vs all) = 120,000 crosstabs. A sensible implementation reduces the number of crosstabs by repackaging as eight grids, five rows by two columns, giving 1500 * 8 = 12000 tables. There are 1500 unique emails. Manually loading

1500 sheets with a unique email, organisation and eight blocks of numbers is not practically feasible, so it's automate or nothing.

The processing steps are

1. Specify eight grid variables from the forty statements, generate and check
2. Assemble list of email, organisation and mailout ID#
3. For each item on the list
    a. Generate eight tables of five scaled mean scores with first column filtered to the respondent's organisation
    b. Pull the tables into the correct sheet positions
    c. Populate the user email and organisation cells
    d. Save as PDF, using <email>_<organisation>_<mailout ID> as the naming convention
4. Spot-check at random until you are sick of it.

The midi-automation sequence stops at the conclusion of step 3. Step 4 is a decision point, the decision being whether or not to notify the client that the deliverables are ready. If yes, then a second automation sequence takes place, conducted by the client, to effect the mailout using MailChimp or similar.

There is no software reason why the email step could not happen within the loop (simply invoke the Outlook APIs for assembling and dispatching an email with attachment), but the great risk there is of committing too soon – of over-automating. What if the lookup list was wrong? A category miscoded somewhere, the Valids filter not applied correctly, or any other of the myriad little typos, glitches or gotchyas? Then you will have the embarrassment of having sent confidential information to potential competitors. *Abandon all hope ye who enter here*, for your respondents will never forgive you.

If contemplating automated bulk emails, make sure you don't get tagged as a spammer – use a commercial entity if necessary.

The risks for this exercise are nearly all mitigated by spot checking the deliverables. Any processing error should be gross and comprehensive. If 50 random deliverables validate against the source data, the chances that there is a problem with the any of the other 1,450 is vanishingly small.

All the scripting for this was done using VB.Net, executed from Visual Studio, coordinating the cross tabulator (Ruby) with Excel, calling their respective APIs:

```
'' 1. EXTERNAL COLLABORATION SCORE
   name = "IV1"
   top = "Benchmark(" & org_code & ";bmcol)"
   side = "IV1(1/5:#10*(cmn))"           '' five rows of code means
   GenTab(name, top, side)               '' Ruby API
   TryTimeOut(CopySub)                   '' Ruby API
   sheet.Cells(30, 15).Select()          '' Excel API
   TryTimeOut(PasteSub)                  '' Excel API
…
…
   '' Excel API to save as PDF
   sheet.ExportAsFixedFormat
    (Microsoft.Office.Interop.Excel.XlFixedFormatType.xlTypePDF,
     targetfilename,
     Microsoft.Office.Interop.Excel.XlFixedFormatQuality.xlQualityStandard)
```

The one difficulty when developing this process was that the spawned Paste to Excel could fail arbitrarily because the table generation is much faster than the transfer, resulting in clipboard conflicts. This was solved by wrapping the Copy and Paste steps in a Timeout function which waits for the clipboard and Excel to catch up. The alternative is to use the OpenXML SDK [25], but most of the time is expended in saving as PDF, so not much advantage, but at the expense of having to do some serious programming (as opposed to mere scripting), opaque to all but OpenXML experts, which has consequences for maintenance. The VB.net fragment above is completely self-documenting as to what is happening at each line, and because interpreted (rather than compiled), the script will stop at a problem line, a great aid for diagnostics.

Always look for checksums, eg count of PDFs = count of valid emails, and automate the checks for absurdities, such as a score > 100 or < 0.

Again, all the advantages are fulfilled, especially scalability, since the difference between 1,500 and 15,000 is simply machine time, linear to the number of PDFs.

Although the generation of the PDF deliverables is a midi-automation, the entire process through to email receipt is really a macro-automation, because there is a decision point, and because multiple discrete machines running multiple applications are required.

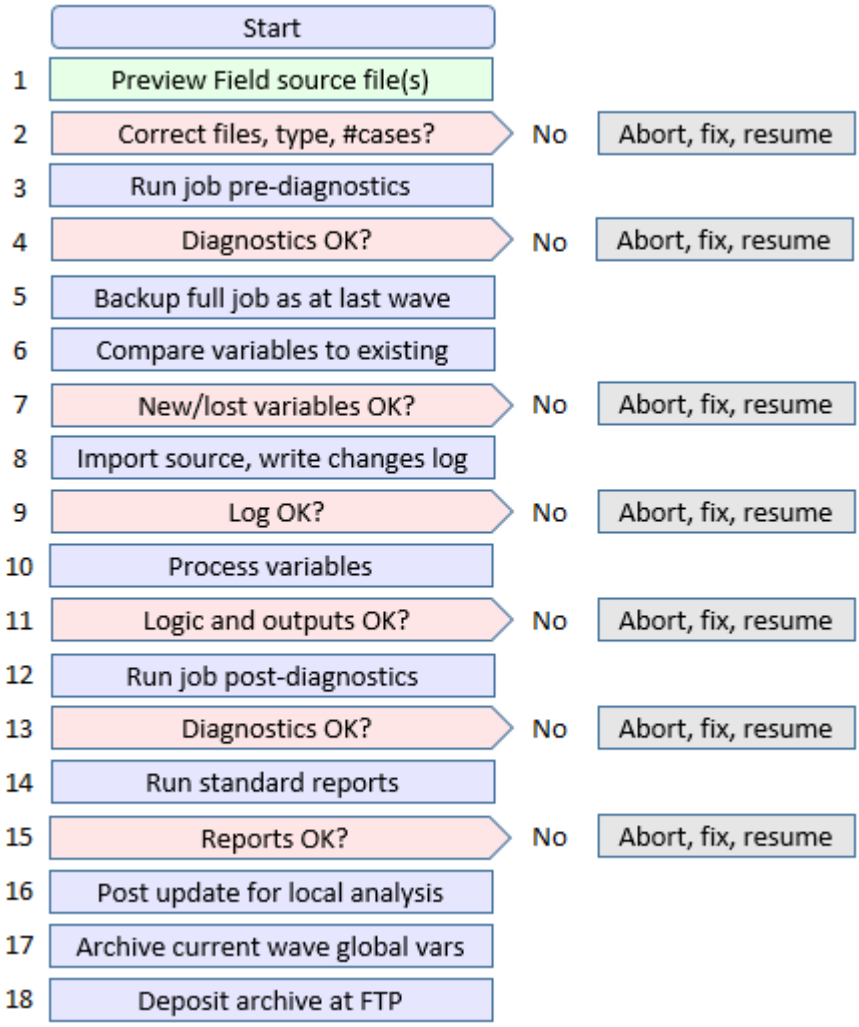
Macro-automation

A macro-automation has at least one and often many decision points which straddle sub-processes at the micro or midi levels.

Automating a Global Tracker

The case study here is an abstraction of a global tracking job, potentially many employees across hundreds of machines.

As a flow chart, the automation steps and decision points (in summary) for a single local job are



These 18 steps are executed against each wave of each local job. Apart from step 1, which can only be done manually, the automatable/scriptable action steps are in blue. The decision points are red. The scripted steps must deliver intermediate processing reports (*confessions*) which the user consults at the decision points. Errors or unexpected inputs/outputs must be attended to (*atoned for*), with the problem identified and remedied before proceeding to the next step. Failure to do so (*a deadly sin*) will guarantee you a reservation somewhere down around the ninth circle.

Step 1, Preview Field, must be done manually – otherwise you could have the wrong files for this job or wave, or missing or even too many cases, or missing coded verbatims, whatever. You do not want to get to step 13 Post-Diagnostics only to discover that you have imported the wrong wave.

Step 3, Pre-Diagnostics, is needed here because the job will have been subjected to active use since the last update, so analyses created by researchers need to be checked and preserved. Also, because of active use, the job may have been damaged in some way – inadvertent deletions, bad syntax, overwriting measures, etc. If your system isolates updatable components from researchers' work, there is still the possibility of machine errors and file corruptions. The only way to be sure the foundation for the current update is sound is to run the general diagnostics first. Typical diagnostics include comparing case data against the previous update to the last common case, data maps showing where variables are in/out of field, spread statistics on weights, check-sums for nets, quota and other base counts, etc.

Step 4, Diagnostics OK, is a decision point. Decision points cannot be automated. A devil's advocate might contend that, for an automated procedure, there are quite a few interruptions, and that is true, but these are at each juncture where a disaster could happen. Tracking jobs are heterogenous and structurally dynamic, so automated rules-based decision-making is not possible, and how to address any issues arising requires a human who understands the system and can identify variations from the expected.

Step 5 - with the job checked, a backup is made as a safety net (never burn your bridges), and for tracing historical issues which may arise in the future.

Step 6, Compare Variables, does an exhaustive comparison by name of the variables currently in the job, versus those about to be imported. The user would expect to see a list of new variables arising from new questions, and a list of retired variables arising from removed questions, and the lists should exactly match expectations for instructed changes at this wave.

Step 8, with all the ducks in a row, the new case data can be imported. All and any changes in variable descriptions and code frames should be reported to the user for confirmation. You do want to see things like new codes, fixed spelling and punctuation, and occasionally an improvement in wording (such changes are usually instructed and hence anticipated). You do not want to see that brand codes have shifted, or rating scales inverted, or a code label or variable description which is obviously wrong (often from copy/paste errors by the questionnaire programmers).

Step 10 is where the source variables are used to create new variables for analysis. This is typically several hundred nets, grids, summaries, banners, quantitative buckets, etc.

Step 12 is the same as step 3, but now performed on the updated variables.

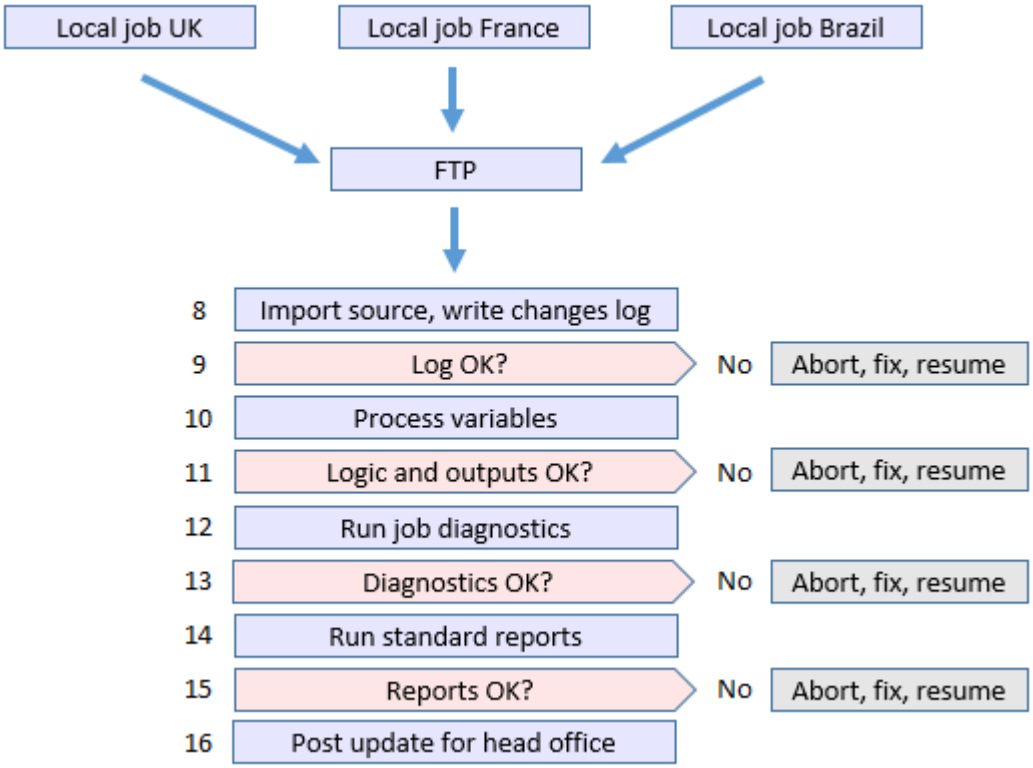
Step 14 updates the standard reporting regime, comprising all tables and charts the researchers expect to be ready-to-go.

Steps 16 to 18 are file movements, which can be easily automated as a single script, but in practice manual supervision of uploads can be safer – depends on how much you trust your network and internet connections.

For step 17, Archive Globals, only useful variables should be sent on. There is usually not much interest in a global analysis of respondent IDs, interview start/stop times, skimmer checks, etc. There may also be variables of local interest only. A global tracking job is big, and gets bigger at each update, so discipline is advised. The archive should comprise new case data only.

Trying to eliminate any of the decision points is over-automating, leading to loss of flexibility to address issues and greatly increasing the risk of reruns. For a more detailed account of the many considerations in tracking job update procedures, see Chant, Automating Continuous Tracking passim [26]

The care taken to ensure a clean job at the local level reduces the steps for merging to the global version, which will comprise all cases from all localities.



If the local jobs are at least nearly always reliable, the decision steps may be optionally skipped, at the risk of an occasional rerun if things go awry. However, when it comes to data processing a tracking job, it is simply not possible to be too paranoid.

Conclusion

Micro-automation is safe, in regular use, and hence beyond contesting. We have voted it a success by wholesale adoption. But midi and macro pose risks. The recommendations here, particularly; identifying the appropriate decision points between automated sub-processes, keeping expectations reasonable, ensuring competent staff and curtailing over-confidence and over-ambition, should substantially increase the probability of keeping the hell fires at bay. You never want to be in the position where you have to say to management or clients:

> *Through me you pass into the city of woe*
> *Through me you pass into eternal pain*
> (Inferno Canto III)

Rather, we want your end clients (who pay our way), in awe of your presentations, to say:

> *In its depths I saw in-gathered, and bound by Love into one volume,*
> *all things that are scattered through the universe, substance and accident*
> *and their relations, as if joined in such a manner that what I speak of is One simplicity of Light.* (Paradiso Canto XXXIII:49-145 The Final Vision)

Notes

1. For the calculations for California, see

https://wattsupwiththat.com/2013/01/02/the-cost-in-human-energy/

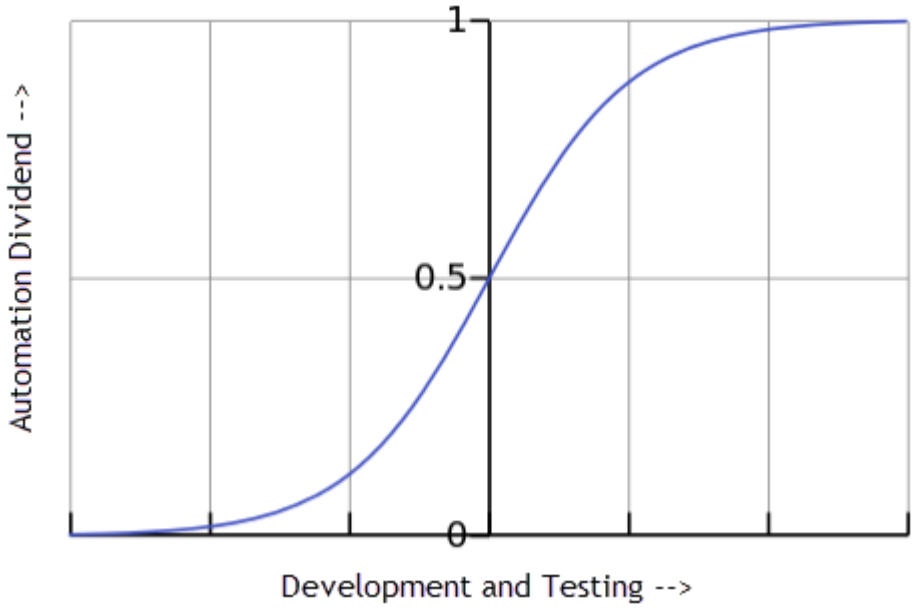
2. Converse pp xx-xxiv. Many others have raised similar questions regarding the potential for diminished data quality from self-selected internet interviews.

3. https://personalmba.com/paradox-of-automation/]

   See also Chant, Automating Continuous Tracking, page 7.

4. Parkinson, C N, Parkinson's Law: The Pursuit of Progress (London, John Murray, 1958)

5. The logistics curve for our purposes is



$$f(x) = \frac{L}{1 + \mathrm{e}^{-k(x - x_0)}}$$

where

- x_0 = the x-value of the sigmoid's midpoint,
- L = the curve's maximum value, and
- k = the steepness of the curve.[1]

(adapted from https://en.wikipedia.org/wiki/Logistic_function)

In short, early to midway development sees great gains, which rapidly level off as the system matures, to the point where no further improvement is possible.

6. Personal efforts to replicate sentiment analyses from first principles have not been successful, see Chant, Exposing and Quantifying Narrative and Thematic Structures, page 34 ff. Natural language processing and text classification has extreme difficulties with everyday rhetorical

constructs such as sarcasm, irony, puns, double entendres, exaggeration, etc. See https://www.theregister.co.uk/2017/04/24/computer_scientists_sarcasm_database/

7. That neural nets are completely opaque is the dirty little secret of AI. A net may appear to be correct, but for the completely wrong reasons. As a simple example, tweeter @XXX makes many negative comments, so a neural net may appear to be classifying negativity well, whereas in fact it is simply identifying a tweeter's handle, multiplied by many retweets. An early example I studied at University was a US military net which supposedly identified tanks. It passed all tests. But on the big demonstration day, it failed completely. As best as could be determined, the training had all been done on cloudy days, and the demonstration on a sunny day, so the net had (we think) mastered only the ability to classify a cloudy day, and 'knew' nothing about tanks. For a nice discussion of neural net opacity, see https://news.ycombinator.com/item?id=9585172

   On text classification issues, see

   https://motherboard.vice.com/en_us/article/ai-can-be-fooled-with-one-misspelled-word

   On vision classification issues, see

   https://motherboard.vice.com/en_us/article/machine-vision-google-adversarial-images

8. https://en.wikipedia.org/wiki/Ostracism

9. Booth, Charles, Labour and Life of the People of East London. Vol 1. London, Williams & Norgate 1889

10. Reproductions of the original forms can be found at https://booth.lse.ac.uk/learn-more/what-was-the-inquiry

11. Booth, Volume 1, page 24

12. Bales, page 434

13. https://www-03.ibm.com/ibm/history/exhibits/vintage/vintage_4506VV2139.html

14. Converse, page 96

15. Converse, page 372

16. The 1950 Censuses: How They Were Taken page 38

17. IBM 101 Electronic Statistical Machine, page 5

18. IBM 101 Electronic Statistical Machine, page 23

19. IBM 101 Electronic Statistical Machine, page 25

20. Converse, page 372

21. Converse, page 373

22. Converse page 373

23. See
    https://www.codeproject.com/Articles/1181809/How-Retailers-Use-Latest-Techniques-from-Machine-L

    https://www.r-bloggers.com/setting-up-the-twitter-r-package-for-text-analytics/

24. Chant, Exposing and Quantifying Narrative and Thematic Structures in Well-formed and Ill-formed Text, pp 16-19

25. Since Office 2007, document files are zipped XML, as opposed to the old binary formats. The XML is fully documented, hence the name OpenXML. This makes it possible for external application software to write MS Office documents as plain text files. The OpenXML SDK expects C#, and the code to achieve the simplest things (like a single coloured letter in a word, a cell border) is long and complicated, requiring software skills well beyond mere scripting.

26. Chant, Automating Continuous Tracking

BIBLIOGRAPHY

1. Bales, Kevin. Early Innovations in Social Research: The Poverty Survey of Charles Booth. Department of Social Science and Administration, London School of Economics and Political Science, University of London.
   http://etheses.lse.ac.uk/55/1/Bales_Early_innovations_in_social_research.pdf

2. Booth, Charles, Labour and Life of the People of East London. Vol 1.
   London:Williams & Norgate 1889

3. Chant, Dale. Automating Continuous Tracking: The Ideal System.
   Delivered at Association for Survey Computing conference:
   Getting the Message Across - Automating and Communication Survey Results
   Imperial College, London, October 2008
   http://redcentresoftware.com/wp-content/uploads/2012/06/Automating-Continuous-Tracking-Sep-09-25-page1.pdf

4. Chant, Dale. Exposing and Quantifying Narrative and Thematic Structures in Well-formed and Ill-formed Text.
   Critical Reflections on Methodology and Technology: Gamification, Text Analysis and Data Visualisation.
   Edited by D Birks et al Compilation ©2013 Association for Survey Computing
   http://redcentresoftware.com/wp-content/uploads/2012/06/Exposing-and-Quantifying-Narrative-and-Thematic-Structures-in-Well-formed-and-Ill-formed-Text_130912.pdf

5. Converse. Jean M. Survey research in the United States: roots and emergence 1890-1960. Originally published: Berkeley : University of California Press. c1987. ISBN 978-1-4I28-0880-4. (obtained by Google Books)

6. Dante, The Divine Comedy

7. Reference Manual, IBM 101 Electronic Statistical Machine, International Business Machines Corporation, 1958

8. U.S. Bureau of the Census, The 1950 Censuses, How They Were Taken, Washington DC, 1955 (free Google eBook)

9. https://en.wikipedia.org/wiki/Automation
   I found this a useful overview of automation in general.

[end of document]

