Cornell University School of Hotel Administration

The Scholarly Commons

Center for Hospitality Research Publications   The Center for Hospitality Research (CHR)

5-24-2017

seniority_list: A Tool to Address the Challenge of Airline Mergers and Labor Integration

Robert Davison
Ruby Data Systems, Inc.

Sean Edmund Rogers Ph.D.
Cornell University School of Hotel Administration, ser265@cornell.edu

Follow this and additional works at: https://scholarship.sha.cornell.edu/chrpubs

Part of the Business Administration, Management, and Operations Commons

seniority_list: A Tool to Address the Challenge of Airline Mergers and Labor Integration

Abstract

Integrating employee groups from separate firms into a combined, well-functioning workforce presents one of the most difficult challenges in a corporate merger. This has particularly been the case in the recent airline mergers in the U.S. that have left three large legacy airlines, namely, American, Delta, and United. Carriers in these mergers have, in some cases, seen years of arbitration and litigation, employee turmoil and labor union decertification, and delays in operational integration and the realization of anticipated merger synergies. In response to this situation, this report introduces seniority_list, a computer-based tool that can be used by unions, employee groups, arbitrators, airlines, and consultants in their workforce integration efforts, analyses, and recommendations. The report demonstrates how the tool addresses such variables as employee tenure, jobs available, and furlough recall schedules, together with ordering and conditions on integration alternatives, to comprehensively assess the short- and long-term impact of workforce integration strategies. The purpose of seniority_list is to help speed up post-merger labor integration, enhance outcome fairness for merged employee groups, reduce conflict, and allow airlines to more quickly realize the operational and financial benefits expected from a merger.

Keywords

airlines, unions, mergers, pilots, labor integration, seniority integration, seniority_list

Disciplines

Business Administration, Management, and Operations

Comments

Required Publisher Statement

The Hotel School

Cornell
SC Johnson College of Business

seniority_list:

A Tool to Address the Challenge of Airline Mergers and Labor Integration

by Robert Davison and Sean E. Rogers

EXECUTIVE SUMMARY

Integrating employee groups from separate firms into a combined, well-functioning workforce presents one of the most difficult challenges in a corporate merger. This has particularly been the case in the recent airline mergers in the U.S. that have left three large legacy airlines, namely, American, Delta, and United. Carriers in these mergers have, in some cases, seen years of arbitration and litigation, employee turmoil and labor union decertification, and delays in operational integration and the realization of anticipated merger synergies. In response to this situation, this report introduces seniority_list, a computer-based tool that can be used by unions, employee groups, arbitrators, airlines, and consultants in their workforce integration efforts, analyses, and recommendations. The report demonstrates how the tool addresses such variables as employee tenure, jobs available, and furlough recall schedules, together with ordering and conditions on integration alternatives, to comprehensively assess the short- and long-term impact of workforce integration strategies. The purpose of seniority_list is to help speed up post-merger labor integration, enhance outcome fairness for merged employee groups, reduce conflict, and allow airlines to more quickly realize the operational and financial benefits expected from a merger.

Robert E. Davison is the founder of Ruby Data Systems, Inc. and is also an Airbus captain for a major US airline. He has flown the A330, B767, B757, A319/320/321, MD80, B737, DC3, SF340, and EMB110 aircraft in 33 years of flying for major and regional carriers. After observing and personally experiencing multiple contentious mergers, Captain Davison recognized the limits of historical workforce integration procedures and developed the seniority_list program. Robert has served pilots and their families on union merger committees for over 5 years, including in the position of Chairman. His technical expertise includes data modeling, analysis, and visualization utilizing various specialized libraries associated with the Python programming language, and advanced proficiency with Microsoft Excel and VBA. Captain Davison is a graduate of Purdue University.

Sean E. Rogers, Ph.D., is an assistant professor at the School of Hotel Administration (SHA) in the Cornell SC Johnson College of Business. Prior to joining SHA, Rogers was the management PhD program director and an assistant professor at New Mexico State University, and before that he was a visiting assistant professor at the

Anderson School of Management at the University of New Mexico.

Before earning his PhD in industrial relations and HR at Rutgers University, Rogers spent several years working in the airline industry and logistics. His corporate experience includes positions in industrial engineering, airline network planning, schedule planning, pricing, and financial analysis at US Airways, America West Airlines, ATA Airlines, AirTran Airways, and UPS. Rogers also spent nine years in the US Army Reserves as a transportation management coordinator and stevedore.

Sean's current research interests include unions and labor-management relations (especially in airlines and higher education), workforce diversity and employment discrimination, and volunteerism. His work has been published in *Industrial and Labor Relations Review, Industrial Relations Journal, Industrial and Organizational Psychology: Perspectives on Science and Practice, Nonprofit and Voluntary Sector Quarterly, Journal of Health Organization and Management*, and *Hospital Topics*. Rogers is a certified Senior Professional in Human Resources (SPHR), and a Society for Human Resource Management Senior Certified Professional (SHRM-SCP

seniority_list:

A Tool to Address the Challenge of Airline Mergers and Labor Integration

by Robert Davison and Sean E. Rogers

n a 1955 article entitled "Seniority Problems in Business Mergers," published in the journal *Industrial and Labor Relations Review*, economist Mark Kahn wrote: "Every industry with seniority practices has encountered seniority integration disputes at least sporadically, while some, especially in transportation, have been plagued by them." [1] Professor Kahn's statement remains no less true today than when he wrote it more than 60 years ago.

---

[1] Kahn, M. L. (1955). Seniority problems in business mergers. Industrial and Labor Relations Review, 8: 362)

Participants in the approximately 50 airline mergers and acquisitions from 1930 through 2013 may have felt that they were in the midst of a plague when it came to integrating their unionized employee groups into a single, merged workforce.[2] Among modern major air carriers, all except Delta Air Lines and JetBlue have significant labor union representation across most "crafts" or "classes" of employees.[3] Thus, airline mergers and acquisitions almost always encounter the seniority integration problems Kahn identified.

These problems have often resulted in multi-year legal disputes, union de-certification, delays in merger implementation, and even new regulations and legislation being enacted to prevent seniority disputes. Classic examples of such disputes include the 1952 Pan American–American Overseas pilots and 1962 United Air Lines–Capital Airlines pilots. As an example of merger-driven regulatory changes, in 1972 the Civil Aeronautics Board (the now-defunct agency then responsible for regulating airline operations, including routes and fares) established the Allegheny-Mohawk Labor Protective Provisions, or LPPs, which arose out of the seniority integration list proceedings from the 1972 merger of Allegheny Airlines and Mohawk Airlines (precursors of US Airways). The LPPs basically required a newly merged airline to "make provisions for the integration of seniority lists in a fair and equitable manner."[4] Amid concerns about the treatment of unionized employees of Trans World Airlines (TWA) following its acquisition by American Airlines in 2001, Missouri senators Claire McCaskill and Christopher "Kit" Bond introduced federal legislation aimed at guaranteeing LPPs for workers affected by merger-induced seniority list integration. The McCaskill-Bond Statute became an amendment to the Federal Aviation Act in 2008.[5]

Introduction to seniority_list

In view of the obdurate challenges relating to seniority-based workforce integration, we developed seniority_list to provide an unbiased, easy-to-understand, robust analytical tool for employee groups, labor unions, airlines, arbitrators, and consultants to use when they are attempting to combine seniority-based work groups. The tool uses readily available employee and company information to model the effects of post-merger, combined seniority

list ordering proposals over a wide range of metrics and conditions, doing so in a manner that is quickly adjustable and can handle multiple iterations. Although it was designed with airline pilot workforce integration in mind, seniority_list can be adapted to any industry or group where workers operate under a seniority system.

The seniority_list tool uses the Python programming language, in conjunction with several Python numerical and data analysis libraries (i.e., pandas, NumPy, matplotlib, and seaborn). Using seniority_list requires a basic level of familiarity with computer language programming, as well as access to the supporting libraries and software programs. The seniority_list program, all needed programming tools, and the user interface are all open-source, and freely available to download, use, and in some cases, modify.[6]

While all of the particulars of seniority_list cannot be described here, this report **(1)** discusses the basic concepts and terminology of seniority_list, **(2)** outlines the process of using seniority_list from start to finish, and **(3)** demonstrates a sample operation, output, and analysis, as well as editing the analysis. Complete instructions for using the tool can be found at the documentation website for seniority_list at rubydatasystems.com. The website contains step-by-step instructions on how to download the tool and all supporting libraries and programs, how to adapt the tool to the particular conditions of a given seniority integration, and conceptual explanations of what the tool attempts to accomplish through its modeling.

Basic Concepts and Terminology of seniority_list

A key attribute of seniority_list is that it is focused on outcomes, rather than on processes alone. A particularly lengthy workforce-integration process, for instance, was that of the pilots in the 2005 America West Airlines–US Airways merger, followed by that airline's 2013 merger with American Airlines. The pilot integration dispute of America West and US Airways was not resolved before that merged airline (then doing business as US Airways) merged with American Airlines. Seniority list integration among those three pilot groups was finally resolved in September 2016, approximately *eleven years* after the initial merger.

In this and the other disputes that we mentioned above, the emphasis has been placed on the *process* of integrating seniority lists rather than the *actual results* of the integration. That is, decision makers have been singularly

[2] See: http://airlines.org/dataset/u-s-airline-mergers-and-acquisitions/

[3] https://nwlaborpress.org/2012/10/fly-the-union-friendly-skies/. Pilots at Delta Air Lines are unionized, but most other large employee groups are not.

[4] http://www.mondaq.com/unitedstates/x/164186/Aviation/Seniority+Integration+And+The+MccaskillBond+Statute

[5] http://naarb.org/proceedings/pdfs/2013-191.PDF

[6] If improvements or other changes are made to seniority_list, the only caveat is that these modifications will also made free and available to the public. Source code for seniority_list is hosted at: https://github.com/rubydatasystems/seniority_list.

concerned with perceived equity and fairness during the process and its immediate outcome. Our horizon involves the results of workforce integration for the working conditions, internal labor market mobility, and lives of employees in the years after the integration has nominally been completed.

To that concern, seniority_list offers a dynamic, analytical approach to understanding the long-term quantitative effects of various alternative workforce integration solutions on the personal and professional lives of unionized workers caught up in a corporate merger.

Seniority

At the heart of any explanation of an airline pilot seniority list and how it operates is one basic idea: once a pilot's position on a seniority list has been set, the advancement and job opportunities available to that employee are directly related to the birth dates of the other pilots senior to him or her on the list.[7] For example, the number of steps a pilot can advance upward on the seniority list over the course of his or her entire career will be capped by the number of more senior pilots with an earlier birth date, because the affected employee will retire prior to those who are younger and more senior to him or her.

Additionally, a pilot's rate of advancement is directly proportional to the ranges of birth dates belonging to senior employees. Someone on the seniority list who is younger than any of their more senior colleagues will move up to the number-one list position at some point. That employee's accession to the number-one position depends on the ranges of senior employees' birth dates. For example, if an individual is on a list where all pilots senior to her will be retiring within the next month, after the next month that pilot will be number one on the list for the remainder of her career. Conversely, if an individual is on a list where all employees senior to him will be retiring two days prior to his own retirement, he will stay right where he is on the list with no advancement until two days prior to his retirement.

Employees working under a seniority system are acutely aware of this situation. Seniority determines nearly every aspect of the employee's work life, from job status to work shift to annual vacation. Employees continually look ahead and calculate when they might be able to attain a particular status. Overall, the seniority system is constant and predictable for employees—until a merger occurs. A merger can completely disrupt the existing systems, with significant changes to workers' careers and challenges to their long-held expectations. This is

---

[7] In the U.S., airline pilots have a mandatory retirement age of 65. Thus, the point at which airline pilots will leave the firm and no longer hold a place on the seniority list is fixed and known.

The following list includes terms specific to the seniority_list tool and airline pilot workgroups, as well as others more generic to the computer languages, software, and programs used to run the tool.

Attribute–a homogeneous measurement stored in a column form, such as list percentage, age, or cumulative pay;

Bidding–a job selection process where individual employees indicate job preferences and receive job assignments using those preferences, beginning with the most senior employee and proceeding through the seniority list top to bottom;

Conditions and restrictions ("fences")–job rights or limits assigned to one or more employee groups intended to reduce equity distortion, normally with defined duration and scope;

Dataframe–a data structure organized with rows and columns which is optimized for data analysis;

Dataset–the final data product of the seniority_list calculation section containing multiple columns of analytical information with one row for each employee for every month of the data model;

Date of hire (doh)–the date when a pilot began work for a company;

Delayed implementation–a delay in time between the "official" merger date and the actual implementation of an integrated seniority list;

Density chart–a chart which displays population distribution and concentration per employee group;

Differential chart–a chart which compares the difference between the same attribute from two calculated datasets;

Editor tool–an interactive display which reveals integrated list inequities and permits corrective list adjustments

eg–employee group;

empkey–a unique employee number assigned to each employee;

Furlough (fur)–a leave from work, normally involuntary due to job reductions;

Inactive–employees who are not occupying or bidding for a position which would otherwise affect the job opportunities for those employees below them on the seniority list;

Integrated–the employee groups merged as one using a combined list order, or the calculated data resulting from the combination;

Long-form–a dataset format containing information for every month for each employee remaining on the list for that month;

Python library/package–an add-on program designed to provide additional specialized capability to Python users, such as web development, server operations, or data science;

Recall–a return to work following a furlough;

Script–a program routine designed to accomplish a specific task;

Short-form–a dataset format containing static list data, without any future progression calculations with length equal to the number of employees;

Skeleton–a long-form pre-calculated dataframe containing employee information which is list order independent;

Squeeze–the movement of a selected section of an employee group within an integrated list using the editor tool; and

Standalone–an independent employee group, or the data calculated for an independent group as if a merger had never occurred.

EXHIBIT 1

seniority_list overview

EXHIBIT 2
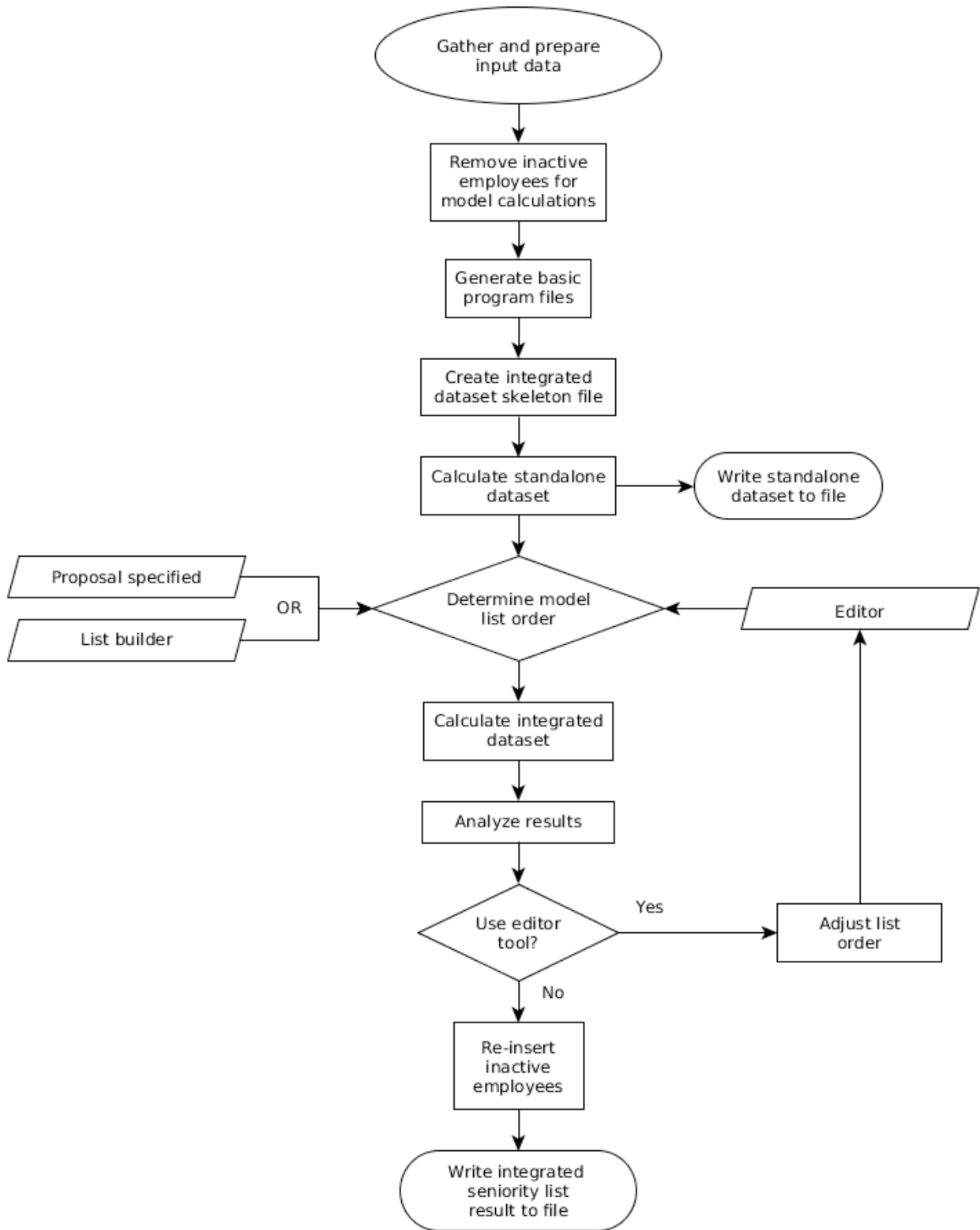
Input file: Employee data (excerpt of master.xlsx)

	A	B	C	D	E	F	G	H	I	J
1	empkey	eg	lname	dob	doh	ldate	sg	fur	line	eg_order
2	10011102	1	tooeyoo	1949-07-13	1973-02-26	1975-01-29	0	0	1	1
3	10010475	1	rubelot	1949-02-05	1975-05-27	1975-05-27	0	0	1	2
4	10013096	1	yeloxid	1949-01-08	1977-01-18	1977-01-18	0	0	1	3
5	10012178	1	xayeaue	1951-06-07	1977-11-15	1977-11-15	0	0	1	4
6	10014447	1	finuceu	1951-10-17	1977-12-09	1977-12-09	0	0	1	5
7	10014384	1	pevagai	1954-06-02	1978-01-12	1978-01-12	0	0	1	6
8	10012843	1	quniy	1953-05-14	1978-02-18	1978-02-18	0	0	1	7
9	10014067	1	qimed	1955-05-25	1978-03-01	1978-03-01	0	0	1	8
10	10010929	1	xovejiq	1955-06-20	1978-03-07	1978-03-07	0	0	1	9
11	10014351	1	yanag	1950-03-22	1978-03-12	1978-03-12	0	0	1	10
12	10011974	1	eipieuv	1955-02-14	1978-03-21	1978-03-21	0	0	1	11

why employee groups fight so hard to preserve status quo whenever seniority lists must be combined.

The seniority_list tool is designed to make this transition as smooth as possible by providing stakeholders and decision makers rapid insight into, and understanding of, the real and practical effect that various proposals would have on employees. It does this by modeling combination outcomes and then measuring and comparing multiple equity factors using specialized algorithms. Additionally, the program is able to build or edit proposals so that job opportunities are balanced and changes to career expectations are minimized for *all* pre-merger work groups.

Assumptions and Boundary Conditions

At some regular interval (e.g., monthly), airline pilots bid on what routes they will fly during some specific period (e.g., the next month). Their rank on a seniority list determines the degrees of freedom of their preferences. The seniority_list tool does not attempt to predict the bidding preferences of individual employees, except to presume that employees will always bid for the highest paying or highest ranked assignment (even though that may not always occur). But it does aim to model the employees' bidding opportunities. For this purpose, the program focuses on variables that are fixed or that can be modeled in a quantifiable state, such as birth dates, jobs available, proposed list orders, furlough-recall schedules, and special conditions or restrictions that may be part of a solution. The overall result of employees' individual choices is a group average, ultimately constrained by list positioning. Thus, seniority_list offers abundantly more insight into the real effect of a potential combined seniority list than has been available to decision-makers in the past.

Because birth dates are fixed, the order of an integrated seniority list is the primary determining factor controlling the careers of pilots within the newly merged airline. Thus, seniority_list was designed to examine the effects of different proposed list orders by creating data models that can extend into the future several months or until the youngest pilot reaches retirement. The tool also determines the best possible job assignments for all employees, as limited by the order of separate and integrated seniority lists. It incorporates other conditional factors such as delayed implementation, job count changes, furloughs, recalls, and special job rights. Many other additional calculated metrics are generated, culminating in a rich set of attributes providing objective, outcome-based analytics for workforce integration decision-makers.

The program does not advocate for any particular starting point or list integration philosophy. It simply calculates results based on the decision-makers' inputs. It can reveal optimal list ordering without the use of any associated conditions and restrictions, or it can offer solutions which contain corrective conditions and restrictions to address equity distortions. An example of that is when list construction is premised on a certain foundational standard, such as order by hire date or ratio, when the native groups possess disparate demographics or job structures leading to an uneven integration.

Procedural Outline of seniority_list

Operational basics. The seniority_list data model is built upon predictable variables, and isolates and treats separately variables that cannot be straightforwardly quantified (see Exhibit 1). We know that certain aspects

EXHIBIT 3

Input file: Compensation data (excerpt of pay_tables.xlsx, 1 of 2 worksheets)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
			1	2	3	4	5	6	7	8	9	10	11	12
1	year	jnum	1	2	3	4	5	6	7	8	9	10	11	12
2	2013.0	1	25.32	120.24	121.21	122.17	123.14	124.10	125.07	126.03	127.00	127.97	128.93	129.90
3	2013.0	2	25.32	102.21	103.02	103.84	104.66	105.48	106.30	107.12	107.94	108.76	109.58	110.40
4	2013.0	3	25.32	94.89	95.65	96.41	97.17	97.93	98.69	99.45	100.21	100.97	101.73	102.49
5	2013.0	4	25.32	60.69	73.18	74.97	76.78	78.61	80.46	82.32	84.84	86.75	88.04	88.70
6	2013.0	5	25.32	51.68	62.27	63.79	65.33	66.88	68.45	70.03	72.17	73.79	74.88	75.44
7	2013.0	6	25.32	48.02	57.85	59.26	60.68	62.12	63.58	65.05	67.03	68.53	69.55	70.06
8	2013.0	7	25.32	64.89	65.41	65.93	66.44	66.96	67.48	67.99	68.51	69.03	69.54	70.06
9	2013.0	8	25.32	33.02	39.71	40.66	41.63	42.61	43.60	44.60	45.94	46.97	47.66	48.01
10	2014.0	1	25.32	129.77	130.81	131.85	132.90	133.94	134.98	136.03	137.07	138.11	139.15	140.20
11	2014.0	2	25.32	110.29	111.17	112.06	112.94	113.83	114.71	115.60	116.48	117.37	118.25	119.14
12	2014.0	3	25.32	102.39	103.21	104.03	104.85	105.68	106.50	107.32	108.14	108.96	109.78	110.60
13	2014.0	4	25.32	65.46	78.95	80.88	82.83	84.81	86.80	88.82	91.53	93.60	94.99	95.70
14	2014.0	5	25.32	55.72	67.16	68.80	70.46	72.14	73.83	75.54	77.85	79.60	80.78	81.38
15	2014.0	6	25.32	51.77	62.39	63.91	65.45	67.00	68.57	70.16	72.30	73.92	75.02	75.58
16	2014.0	7	25.32	69.99	70.55	71.11	71.67	72.23	72.78	73.34	73.90	74.46	75.02	75.57
17	2014.0	8	25.32	35.57	42.79	43.82	44.87	45.93	46.99	48.07	49.53	50.63	51.38	51.76
18	2014.1	1	154.36	155.61	156.87	158.12	159.37	160.62	161.88	163.13	164.38	165.63	166.88	168.13
19	2014.1	2	130.28	131.38	132.44	133.49	134.62	135.66	136.66	137.78	138.75	140.23	141.72	143.18
20	2014.1	3	123.46	124.46	125.46	126.48	127.51	128.52	129.53	130.55	131.55	132.64	133.75	134.84
21	2014.1	4	45.18	83.78	97.86	100.20	102.57	105.13	108.03	110.48	111.66	113.16	114.18	115.20
22	2014.1	5	45.18	70.82	82.68	84.66	86.70	88.85	91.26	93.37	94.32	95.86	97.02	98.15
23	2014.1	6	45.18	67.12	78.35	80.23	82.15	84.19	86.52	88.50	89.44	90.69	91.58	92.46
24	2014.1	7	81.50	82.11	82.78	83.44	84.06	84.74	85.38	86.05	86.70	87.37	88.04	88.68
25	2014.1	8	45.18	45.18	51.84	53.07	54.30	55.65	57.17	58.46	59.07	59.87	60.42	60.93
26	2015.0	1	158.96	160.24	161.54	162.83	164.12	165.41	166.70	167.99	169.28	170.56	171.85	173.14
27	2015.0	2	134.16	135.28	136.38	137.46	138.63	139.69	140.73	141.88	142.88	144.40	145.94	147.44
28	2015.0	3	127.13	128.16	129.19	130.24	131.30	132.34	133.38	134.43	135.47	136.59	137.72	138.85
29	2015.0	4	46.50	86.27	100.76	103.17	105.61	108.25	111.23	113.77	114.97	116.52	117.57	118.62
30	2015.0	5	46.50	72.91	85.12	87.16	89.27	91.48	93.97	96.14	97.11	98.70	99.89	101.07
31	2015.0	6	46.50	69.10	80.66	82.60	84.58	86.68	89.08	91.12	92.09	93.38	94.29	95.20

or parameters are known and unlikely to change (e.g., job counts, retirement counts, and pay scales), while others are likely or sure to change (e.g., individual bidding choices and future employee work leaves). The factors that are known or predictable are incorporated within the model calculations. The unpredictable factors are handled equally for each group so that one can control for their threat to the accuracy of the model. With the effect of the unpredictable variables removed, the results of the calculations will be directly related to the predictable variable inputs. The primary predictable variable is list order, which has by far the most influence on the resultant datasets.

seniority_list constructs a job-level hierarchy in accordance with compensation scales, again assuming that all employees will continuously bid for the highest paying job. Consequently, the resultant employee career metrics reflect and focus primarily on the true effect of the ordering of proposed integrated lists.

Input Data

To get started, log into the seniority_list documentation site, rubydatasystems.com. seniority_list reads input data from four user-formatted Excel files (as shown in Exhibits 2 through 5). The input files contain employee data, compensation information, proposed list orderings, job information, and program settings. File setup is simplified by the inclusion of example templates and extensive guidance from the program documentation.

Input files relating to specific case studies are stored in separate folders, permitting the simultaneous existence of multiple case studies. Switching between case studies for analysis is simple, involving a change of a single program input.

List order proposals submitted by parties are normally read and stored from one of the Excel input files, but new list ordering also may be generated using the functions within the *list_builder* script, or an existing list ordering may be modified through the interactive list editor tool.

Generating Basic Program and "Skeleton" Files

seniority_list begins by reading the input data and preparing certain files needed by the program for dataset generation and other operations. Datasets are built from a basic structure known as the skeleton file, which is a long-form pandas dataframe assembled from one of the Excel input files. The skeleton file contains general employee information which is integrated-list-order independent, together with pre-calculated data common to all subsequent dataset production. Only the order of the skeleton file is adjusted for different integrated list proposals.

Input file: Proposed integrated list ordering (excerpt of proposals.xlsx, 1 of 3 worksheets)

	A	B
1	order	empkey
2	1	10011102
3	2	10010475
4	3	10013096
5	4	10012178
6	5	10014447
7	6	10014384
8	7	10012843
9	8	10014067
10	9	10010929
11	10	10014351
12	11	20010692
13	12	10011974
14	13	10014916
15	14	10013233
16	15	10010598
17	16	10014294
18	17	20011034
19	18	10011916
20	19	10011396
21	20	10012952
22	21	10013060
23	22	10014377
24	23	10010054
25	24	10012113
26	25	10013413
27	26	10014156
28	27	10011656

Exhibit 5

Input file: Job changes (excerpt of settings.xlsx, 1 of 13 worksheets)

	A	B	C	D	E	F	G	H	I	J	K
1	change	job	month_start	month_end	total_change	eg1	eg2	eg3			
2	jc1	1	35	64	43	40	3	0			
3	jc2	4	35	64	72	66	6	0			
4	jc3	2	1	52	-408	-377	-23	-8			
5	jc4	5	1	52	-510	-474	-26	-10			
6	jc5	3	1	61	411	376	26	9			
7	jc6	6	1	61	411	376	26	9			
8											
9											

scalars | attribute_dict | ret_incr | job_counts | **job_changes** | recall | prex | ratio_cond | ratio_count_cap

Calculating Standalone Datasets

Standalone datasets contain career progression data for each employee group separate from other group(s) as if no merger occurred. Standalone data are useful for comparisons between each other and with integrated datasets. Information pertaining to each separate group may be extracted from the skeleton file quite easily and processed independently. The standalone dataset calculations consider any pre-existing special rights to jobs within one or more of the employee groups, job count changes, and furlough and recall schedules, among other options.

Calculating Integrated Order-dependent Datasets

The production of an integrated dataset is more complex than the standalone datasets, as depicted in the flowchart in Exhibit 6.

A standard provision when integrating a workforce is that employees will be able to keep the job they held prior to a merger, even if the integrated list places a particular employee in a position that they would not otherwise hold. This provision is known as "no bump, no flush." Quite often, due to differences in demographics, hiring patterns, and job opportunities, certain conditions and restrictions (also known as "fences") are applied prospectively to the operation of a combined seniority list. These fences may place a cap or floor on the number of jobs which may be held by employees from one or more of the original groups, provide some sort of ratio assignment process, or apply some other corrective action to ensure an equitable outcome. It is also common to see a time span between the "official" merger date and the actual implementation of an integrated seniority list. This delayed implementation affects the operation of the list.

When calculating integrated datasets, seniority_list is able to incorporate all of the above factors along with other factors such as pre-existing job assignment conditions, job count changes, furlough and recall schedules, and compensation schedules.

Job Assignment within Dataset Generation

Job assignments cascade from top to bottom within each month, in the following manner. Beginning with the highest level job, employees already holding that job from the previous month are assigned. Then, if there are unassigned jobs remaining within that job level and there are special job assignment conditions associated with that job level for the current month, jobs are assigned according to that condition. Finally, any remaining unassigned jobs are distributed to the most senior unassigned workers. The program then proceeds to the next job level for the current month. When all of the job levels for that month are processed, any remaining unassigned workers are marked as furloughed. The assignment routine then proceeds to the next month, repeating until all months have been completed.

Analyzing Results

Once an integrated, order-dependent dataset is generated, the dataset will likely be quite large, as it contains one row for each employee for every month within the model. While the exact size depends on the demographics of the employees, an initial list containing 12,000 employees will produce a dataset with over 1.5 million rows and 35 columns of data.[8]

_____

[8] Despite the file size, this will run on a typical contemporary laptop computer.

EXHIBIT 6

Process for generating an integrated order-dependent dataset

List Order Source
- Editor
- OR
- List Builder
- OR
- Proposal Specified

Gather initial job counts per employee group

Read Skeleton

Read List Order

Read Master List

Gather projected job count changes over time (per job, per group)

Sort skeleton file by month and list order (use editor list order if in "edit_mode") This forms the structural foundation for the overall dataset production

Assign original jobs to each employee group using group job counts, incorporating any pre-existing group job assignment conditions

delayed implementation?

YES → Copy standalone data (from calculated standalone datasets) for the pre-implementation months and carry forward the last month of standalone data for start of integration calculations

NO

Calculate the number of jobs available within each job category for all months (from job changes data input)

Assign jobs to combined group, allowing for delayed implementation, no bump/no flush, assignment conditions, enhanced/basic job levels, jobs available, compensation, and furlough and recall options

After jobs for each month have been assigned, calculate many other attribute columns and add to the dataset

Add compensation data to dataset based on current year, job level, and longevity

Store calculated dataset

EXHIBIT 7

Editor tool process

Once the datasets have been produced, the user is able to explore them using many of the built-in methods of Python and the "scientific stack" libraries (e.g., pandas, NumPy). The datasets may be converted to other types of files for analysis within other programs if desired. Interactive exploration and visualization of the dataset are available through the Jupyter Notebook interface.[9] This allows seniority_list users to easily and quickly transform model output into charts and graphs that depict the effects of integration on the various workgroups.

Within seniority_list there are many built-in plotting functions available to visually explore and contrast

___

[9] (http://jupyter.org/) As stated on its website, the Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text. Uses include data cleaning and transformation, numerical simulation, statistical modeling, and machine learning.

multiple attributes of the datasets. Most of these functions accept a variety of inputs allowing a wide range of analysis. The built-in charts are produced by a Python library called matplotlib and another called seaborn, which is a charting library built on top of matplotlib with a focus on statistics.

Modifying List Order with the Editor Tool (If Needed)

Analysis of the initial integrated dataset will likely reveal certain issues of inequity related to a particular list order proposal. The editor tool can remedy this by allowing adjustment of proposed list order through an interactive process, as depicted in Exhibit 7 and the screenshot in Exhibit 8. The editor tool interface consists of a differential chart, input controls, and a distribution density chart, all of which are described on the following pages.

Exhibit 8

Editor tool interface: Controls, differential chart, and density chart
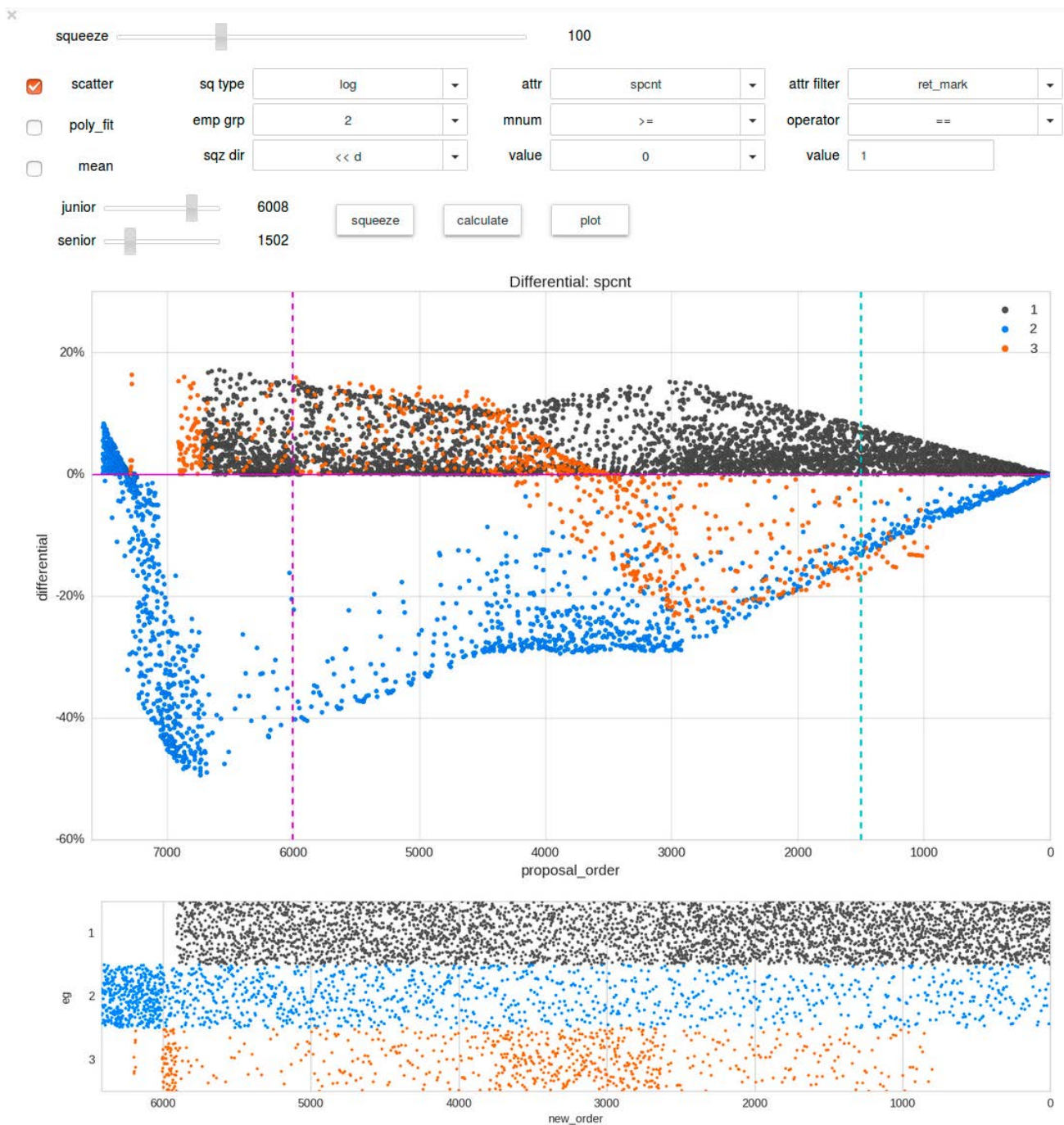
The differential chart is used to quickly reveal outcome-based equity distortions existing within an integrated dataset and to identify where modification of list order may be necessary. The user assigns an attribute for comparison by selecting from a dropdown box containing various dataset measures. Further filtering is available to limit displayed results to a particular month, group, or other attribute. A horizontal slice of the differential chart display may be selected by using two interactive slider controls which position two vertical lines on the chart. The area of the chart between the lines represents a section of the integrated seniority list.

To correct an equity distortion, a user positions the vertical lines on either side of the distortion. An algo-

EXHIBIT 9

RUN_SCRIPTS Jupyter Notebook

EXHIBIT 10

Setting the case study

note: add case study name after script name

```
In [ ]:  %%time
         %run build_program_files sample3
```

rithm within the editor tool then slides or "squeezes" the members from one of the original employee groups up or down the list within the selected section, creating a new, modified order, while maintaining proper relative ordering within each employee group. Following the squeeze operation, a new horizontal list-density chart appears below the main differential chart, representing the new population density produced by the squeeze. The squeeze process may be repeated with different inputs prior to any calculations if the density chart indicates an undesired population shift. Once the user is satisfied with the squeeze, the squeeze-modified integrated list order is then sent back to the dataset creation routine by clicking the "calculate" button. A new dataset is generated and the resultant outcome is displayed on the differential chart for re-examination.

The edit process may be repeated many times, until the observed equity distortion(s) are eliminated or reduced to the extent possible.

Reinserting Inactive Employees

Inactive pilots are defined as those who are not occupying or bidding for a position which would otherwise affect the job opportunities for those employees below them on the seniority list. Examples of inactive employees might include those with a status of medical, military, or supervisory leave. Because inactives do not bid for jobs and have no effect on the operation of a seniority list, they are removed from the list prior to the dataset calculation process. While many on inactive status will return to active status over time, an assumption is made that other employees will do the opposite and provide a counterbalance.

Once a final integrated list order has been determined for all active employees, the inactive employees must be reinserted into the overall list, using the *join_inactives* script. The inactives may be inserted into the integrated list by locating them next to an employee from their native group who is either just senior or just junior to them.

An Example of seniority_list Operation, Output, Analyses, and Editing

Let's examine an abbreviated example of how to use the seniority_list tool to support workforce integration decision making. This example mirrors a recent airline pilot integration attempt in which pilots representing three separate airlines were being integrated as a result of a series of mergers and acquisitions. Anyone can conduct this demonstration analysis, as the needed sample files are included in the downloaded tool. The sample case study is named "sample3" and includes a list of approximately 7,500 pilots from three separate employee groups.

We designed the seniority_list program to use the Jupyter Notebook as its working interface, because this application is especially adept and capable in this role, to include the interactive manipulation and visualization of large datasets and output. Users can also choose to run seniority_list via the command line or analyze the calculated datasets with any suitable program. Again, we do not explain every aspect of the programming below, as comprehensive documentation is available on the seniority_list website.

Starting the Program and Inputting Data Files

Start by opening a Jupyter Notebook window, which will display cells of programming code what can be run independently, modified, and rerun with interactive results occurring in real time. This code can direct seniority_list to perform the actions needed to merge the sample employee lists found in the "sample3" case-study files included with the program. The Jupyter Notebook known as "RUN_SCRIPTS" contains commands and other code used to generate all necessary program files from the sample data and to compute datasets using various proposed list orderings and associated conditions, as shown in Exhibit 9.

Many of the RUN_SCRIPTS notebook script commands accept arguments, or variable inputs, which direct the program to use specific source files for calculation or

EXHIBIT 11

Sample of a plotting notebook (output)
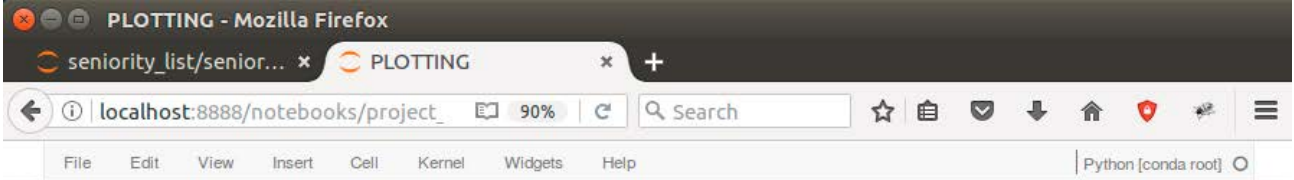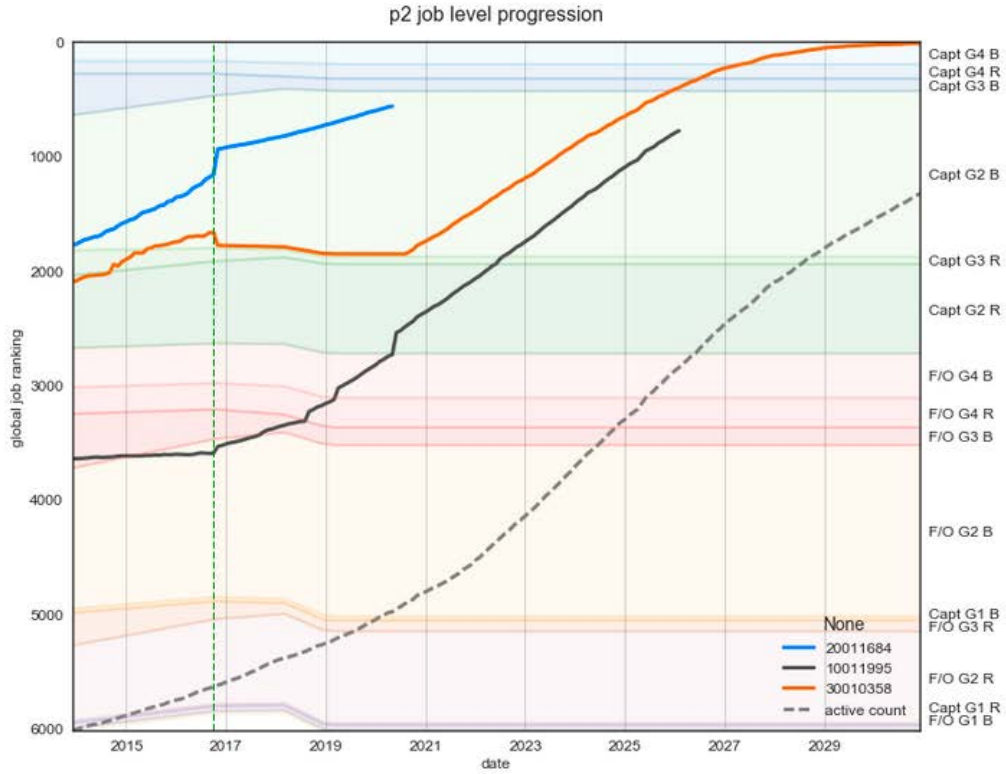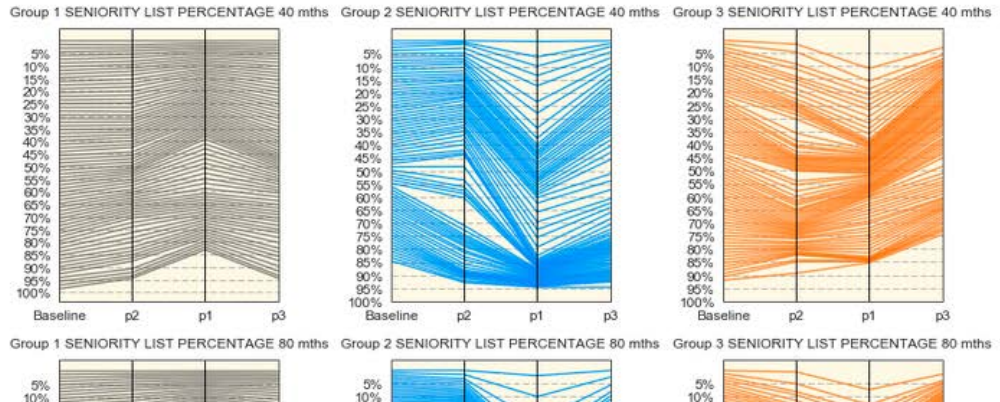
to identify options to be included within the calculations. In fact, seniority_list is directed to the proper case-specific input files by a single argument as an addendum to the "build_program_files" script code line. Exhibit 10 shows the argument ("sample3") set to read the sample3 input files.

Another example of script arguments is found within the last code cell visible in Exhibit 9. That code commands seniority_list to compute a dataset for proposal 2 ("p2") while incorporating a pre-existing job assignment quota condition ("prex") and prospectively applying another job assignment condition ("count"). The referenced conditions are defined by case-specific input file settings and are applied automatically with specialized program functions.

For reference, this process created 20 files, with a total size of 1.2gb, requiring 13 seconds on a Linux desktop machine with an i7 processor, 32gb of RAM, and a solid-state drive.

Sample Output and Explanation

Once the datasets have been produced, the exercise becomes one of examination, visualization, comparison, and potential modification. seniority_list includes many built-in plotting functions designed to be particularly applicable to seniority integration model visualization, but users are free to explore and add to these methods without restriction.

Jupyter Notebook can display charts inline, just below the plotting command code cells. The screenshot in Exhibit 11 is an excerpt of the sample plotting notebook, after running it with a sample dataset calculation output. The inputs within the code cells (function variables) may be changed to view chart output for different attribute measurements or population segments.

Due to size restrictions, the sample output in this introductory report is limited in scope. The sample model comprises sixteen job levels, each representing an airline pilot job at a major airline, in one of four groups based on plane types to which the pilots are assigned.[10] The

_____

[10] In our example, these 16 jobs represent aircraft captains (CAPT) or first officers (F/O). They are currently assigned to fly 1 of 4 groups of aircraft. Pilot group 4 (G4) consists of large wide-body jets, such as the Boeing 787 or Airbus A330. Pilot group 3 (G3) includes small wide-body and large narrow-body jets such as the Boeing 767 and 757. Pilot group 2 (G2) includes narrow-body equipment such as the McDonnell-Douglas MD80, Boeing 737, or Airbus 319/320/321. Pilot group 1 includes small narrow-body or large regional jets such as the Embraer E190.

Additionally, pilots are either in a blockholder (B) or reserve (R) job status. A blockholder is a pilot who has a monthly schedule of trips ahead of time (most desirable for pilots), while a reserve pilot is scheduled to be on call for certain days in that month. Blockholders get

analysis also takes into account whether a pilot has an assigned work schedule (blockholder, or B) or is on standby (reserve, or R). Generally speaking, the captain jobs on the largest aircraft are the highest ranked and best paying positions, and the lowest ranked jobs are first officer positions on the smallest aircraft. There are other factors (which are customizable within seniority_list), but this is the general rule.

Job Transfer

Combining two or more seniority lists will likely result in some shifting or transferring of job opportunities from one employee group to another. This transfer could be to the group's benefit (e.g., a gain in the number of Capt G3R jobs shifted from Capt G2B jobs) or detriment (e.g., a quantity of F/O G3B jobs traded for F/O G2R positions). Jobs available and assigned to members of each employee group within different list order data models may be compared over time. Transfer of jobs by job level (both upward and downward) throughout the model may be calculated and visualized. While the number of jobs in each category remains the same, the employees occupying those positions can be different due to seniority job bidding and awards based on integrated list employee positioning.

The chart in Exhibit 12 displays the change of job assignments over time to one employee group assuming a certain proposed integrated list order that also includes job assignment conditions. In this case, the comparison is between standalone job projections vs. integrated job progressions. The chart legend on the right is arranged so that the most desirable job assignments are at the top, meaning that the jobs in blue are the best (starting with light blue and then descending to dark blue, in the blue group), followed by the green group, and so on. The chart has data presented in a mirror format meaning that the area above the zero line (job transfer gains) is always equal to the area below the line (job transfer losses). The user can quickly see how the job assignments for the selected employee group are affected by the proposed list order. In this case, the jobs above the line traded for jobs below the line are all in a higher category, meaning

_____

more paid working hours per month (approximately 85) compared to reserve pilots who can receive 10 fewer working hours per month (and sometimes not even that). Also included in our output are furloughed (FUR) pilots.

Per the nomenclature above, for example, "Capt G4B" stands for the job of a blockholder captain on a large wide-body jet. In our example output graphs and charts, these jobs are rank-ordered by compensation (from highest to lowest) with Capt G4B being the highest and F/O G1R (a reserve first officer on a small jet) being the lowest of the active pilots, followed by furloughed pilots.

EXHIBIT 12

Output plot depicting the job transfer effects of workforce integration on sample employee group 1
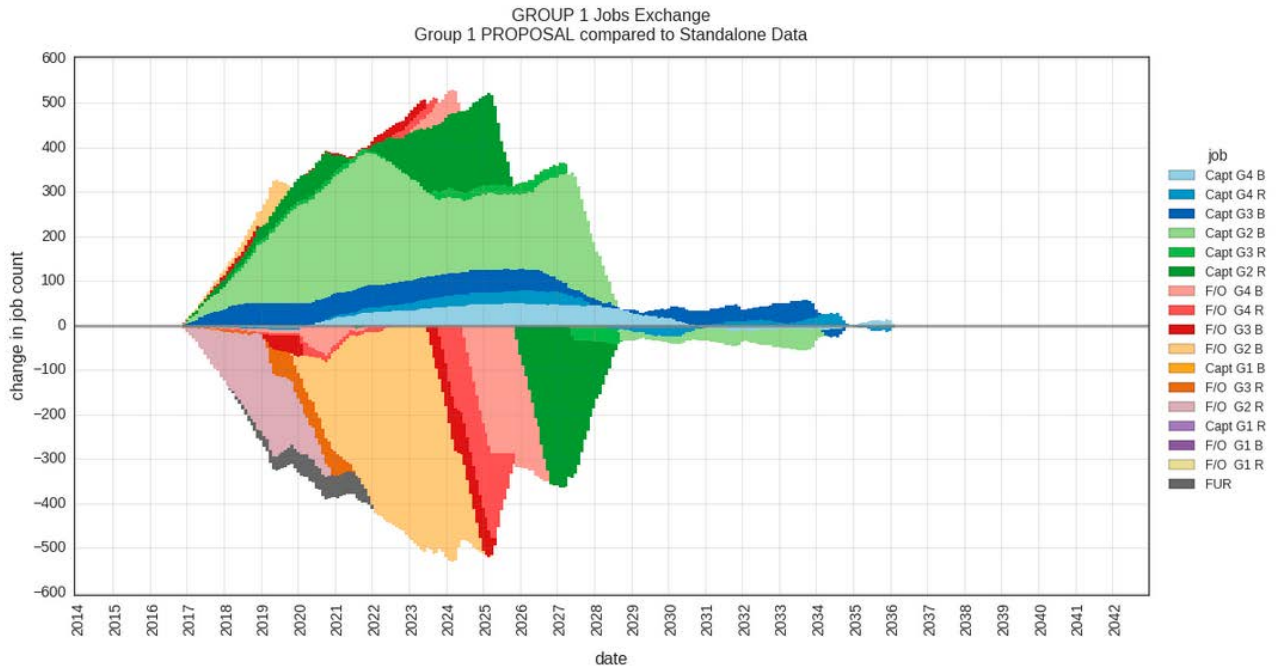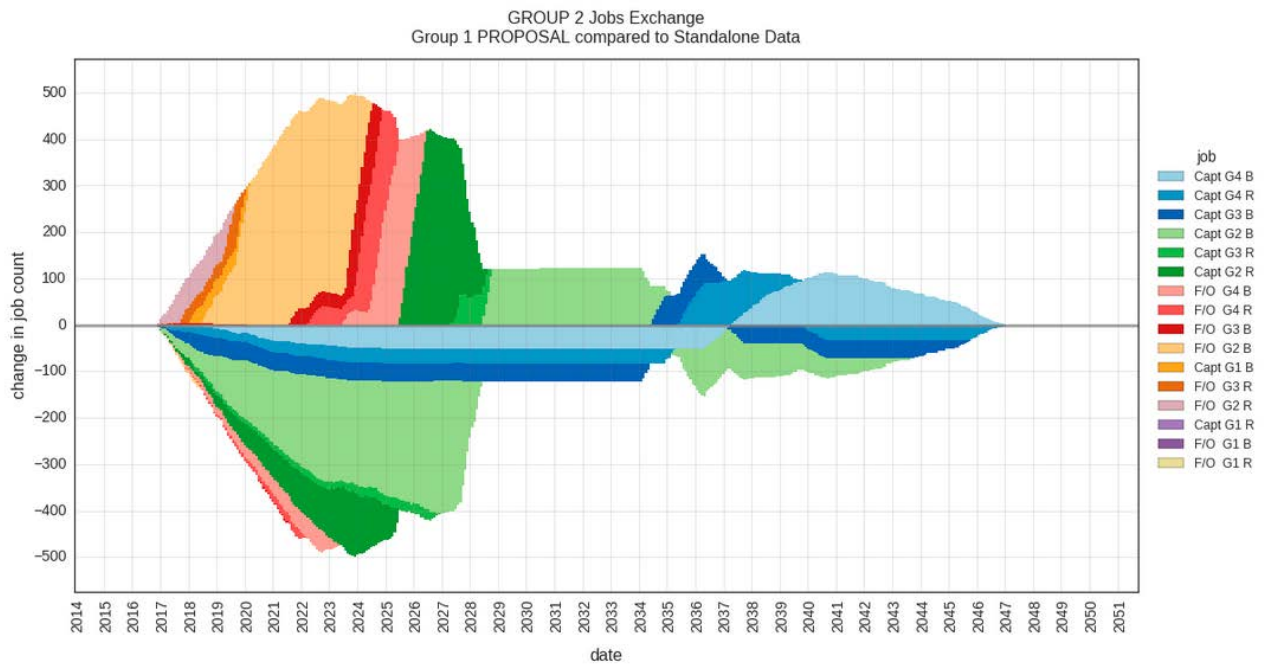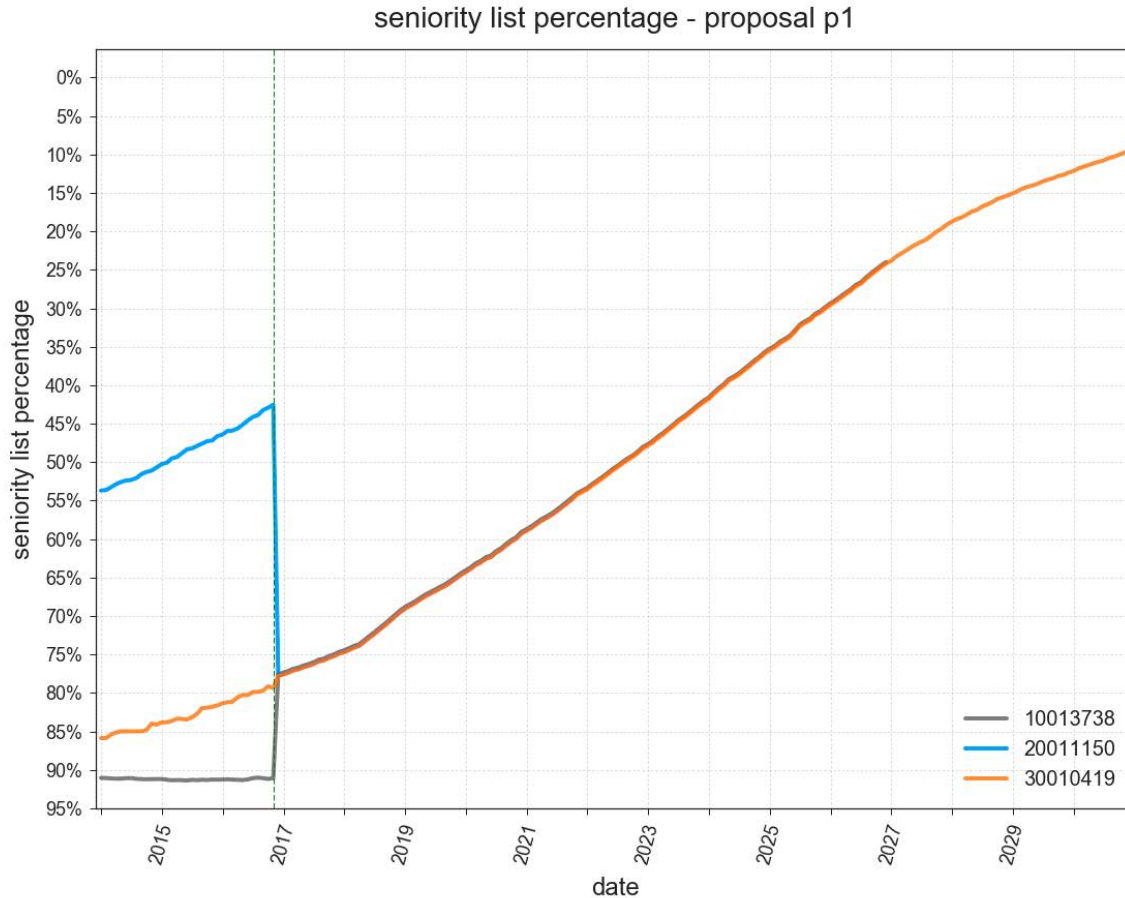

GROUP 1 Jobs Exchange
Group 1 PROPOSAL compared to Standalone Data

EXHIBIT 13

Output plot depicting the job transfer effects of workforce integration on sample employee group 2

GROUP 2 Jobs Exchange
Group 1 PROPOSAL compared to Standalone Data

EXHIBIT 14

Sample list percentage progression for 3 pilots, 1 from each pre-merger airline

this proposal is good for this particular employee group ("employee group 1" in our example).

Because the number of jobs available at any particular time is fixed, the aforementioned accumulation of better jobs to pilots in employee group 1 comes at the expense of another group. Exhibit 13 reveals the effect of the same proposal (from group 1) on the pilots of employee group 2. The short- and long-term effect of the same workforce integration decision on employee group 2 is almost the reverse of that for employee group 1, with losses in every job category. Although we don't show the analysis depicting outcomes for employee group 3, the process is the same as for employee groups 1 and 2.

The samples above compare an integrated dataset to the standalone dataset. The program is also able to compare two integrated datasets; that is, two proposals can be compared side-by-side to visualize how each affects the different employee groups.

An important contribution of the seniority_list tool is that it allows decision makers to clearly visualize the long-term effects of integrated list proposals on all pre-merger workgroups. This information can help remove some of the subjectivity inherent in merger workforce combinations, and lead to smoother and quicker integration.

Modeling Conditional Job Assignment and Fences

The application of "no bump, no flush" and other conditional job assignment rules can have a dramatic impact on individual career progressions and group metrics. This is demonstrated in Exhibit 14, which displays career progression in terms of list percentage for three employees, each from a different employee group and identified by sample employee numbers. List percentage reflects the proportion of pilots who rank above them in seniority. On the example proposed integrated list, which has a

Exhibit 15

Actual job progression of the 3 pilots in exhibit 13, accounting for job assignment conditions and fences
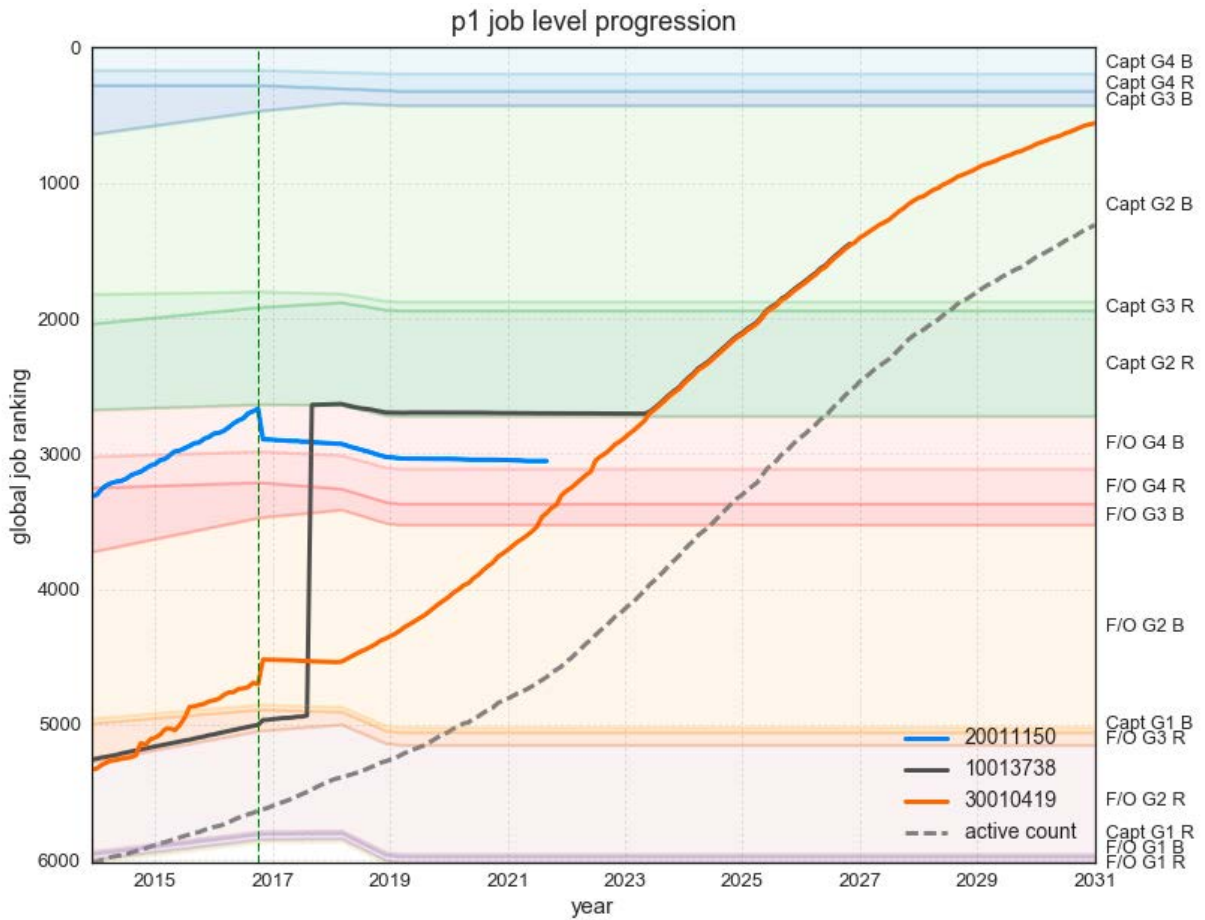
projected implementation date of late 2016, the employees were placed close to each other at approximately 80 percent. After the lists have been combined, the plotted results for the three employees are nearly identical. Over time, older pilots above them on the seniority list retire, and they move together on a percentage basis.

Initially, it appears that the three employees will experience similar job opportunities post-integration. However, due to jobs held prior to the integration, conditional job assignment provisions, and changes in job level counts, the actual jobs held by the three employees will be vastly different from each other. Exhibit 15 reveals a more accurate model of what would occur in terms of jobs available to these three employees. The thin vertical dashed line represents a modeled implementation date. In an airline merger, each group operates independently until the implementation date. The black line represents a pilot belonging to a subset group with rights to a pre-existing special job assignment condition containing quotas

to certain premium jobs, allowing the large jump in job levels to the reserved positions. The blue line represents another pilot who holds a higher ranked job at implementation, which is protected until his retirement. Employees holding a job due to no bump, no flush protection or special job assignment rights remain in that job until their list partners from other group(s) "catch up." Only until the three pilots have reached the point in time where the same bidding opportunities exist for all three do they then move together in terms of list percentage correlating to job assignments, and that occurs only if they have not already retired by that point.

seniority_list's ability to integrate into the model employee-level nuances in job conditions and fences (which may have significant ripple effect on the careers of other employees) is an important and novel contribution. These conditions shape the outcome of workforce integration, and must be taken into account if decision makers are to arrive at the most equitable solution.

Exhibit 16

Effect of integrated list proposal on retirement list percentage, by employee group
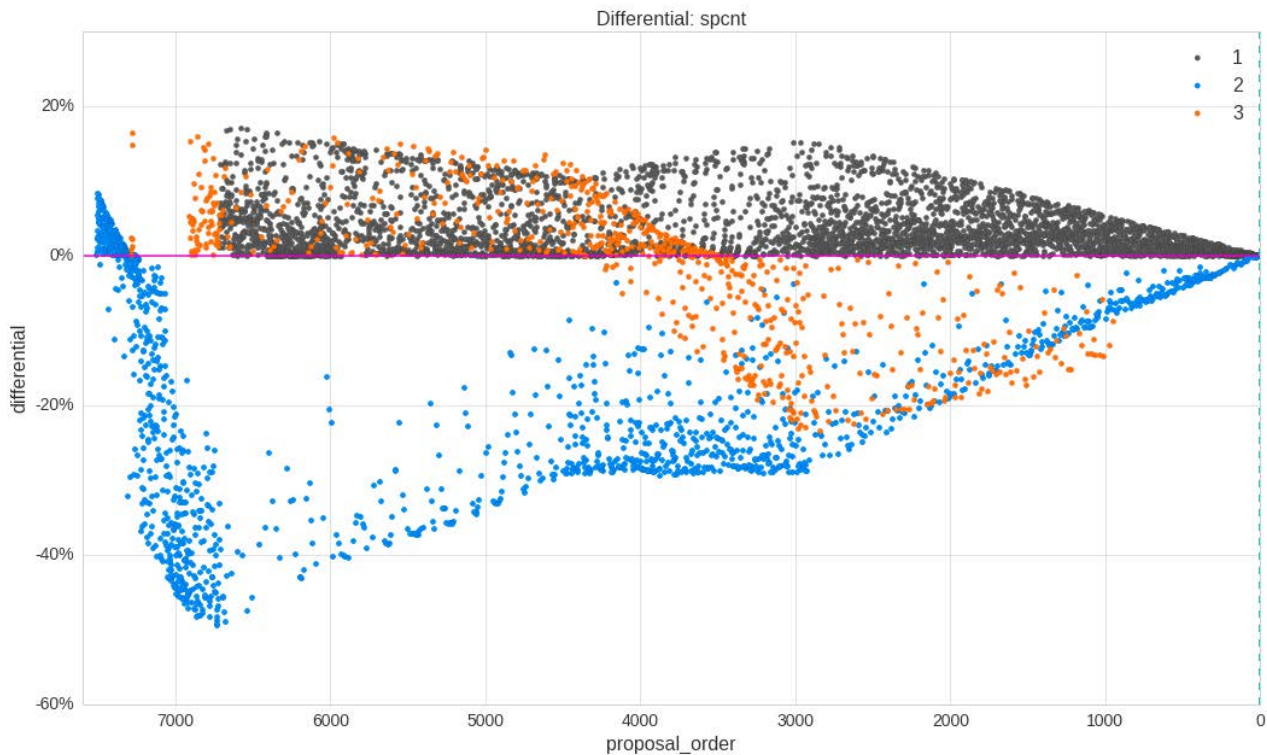
List Modification with the Editor Tool

As discussed earlier, the editor tool shows differential information from calculated datasets resulting from separate list order proposals. The user selects a dataset attribute to compare. In this example, the editor tool output is set to display the difference in employee list percentage at retirement between standalone data and data for an integrated list proposal. Each of the three employee groups is represented by a different color, and each dot represents an individual employee. The y-axis displays the attribute differential (e.g., list percentage or job assignment), while the x-axis displays integrated list order with more seniority to the right and less seniority to the left.

The results in Exhibit 16 indicate that the employee group represented by the black dots will fare much better under the current proposal than it would have with standalone projections. Nearly all of the blue group will do much worse, and the orange group is split.

The output distortions of the initial differential comparison (in Exhibit 16) can be reduced with the editor tool, as shown in Exhibit 17. The list editor can be used

to modify the integrated list order, repositioning the groups or sections of groups up or down the integrated list to remove gains or losses indicated on the differential chart. Each time a modification is made, a new dataset is calculated and a new differential chart is displayed with the updated results. This process only changes the relative order *between* groups; relative list order *within* groups is maintained.

If the differential chart indicates that a portion of one group is experiencing a windfall as compared to its neighbors from other group(s), this is an indication that members of that group should be moved to be made less senior, allowing the other groups to capture positions otherwise held by that group within the data model. A windfall could be indicated by much better average jobs held at retirement or an increase in career pay for one group while another group suffers a loss in both categories. The converse is true for sections of groups negatively affected; they would be moved to more senior positions. The goal of the process is to fine-tune the integrated list order to one which promotes the most equitable outcome possible for *all* of the employee groups.

Exhibit **17**

Edited effect of integrated list proposal on retirement list percentage, by employee group; more equitable solution compared to results in Exhibit 16
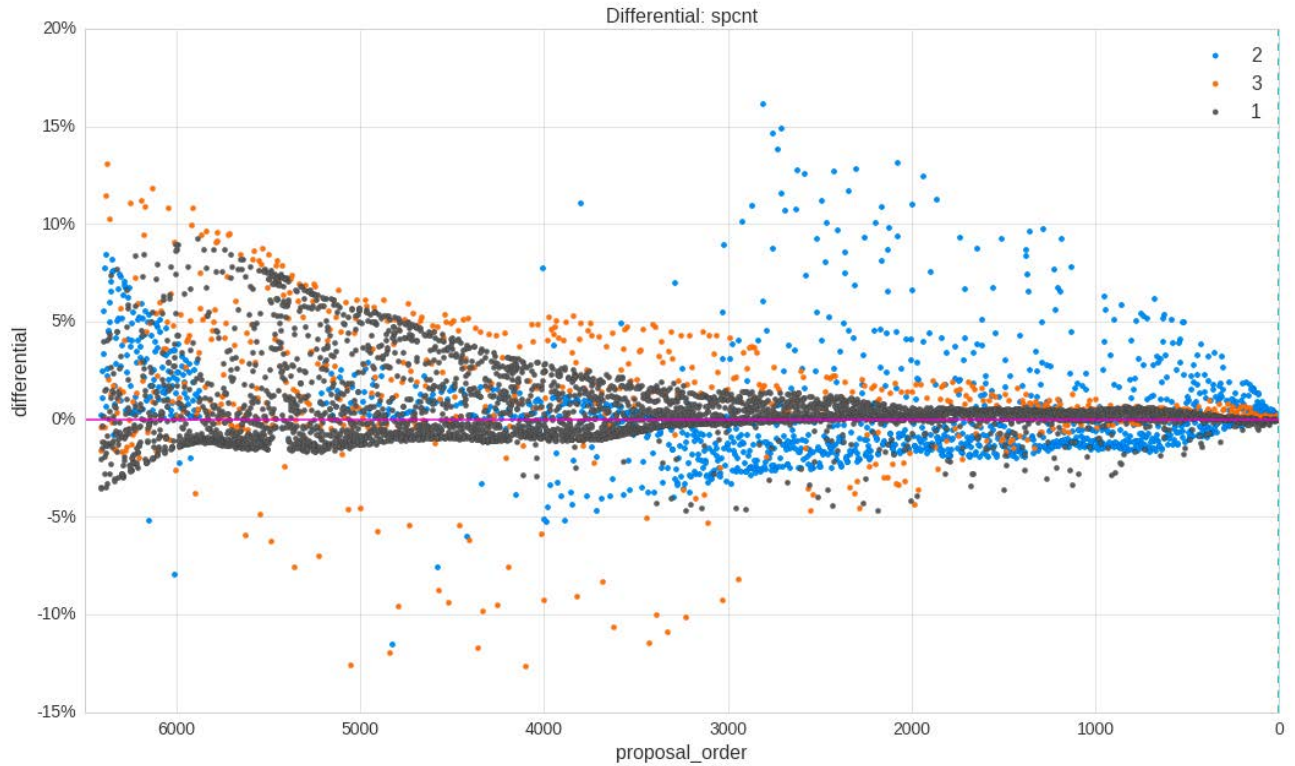
In this case, the first step was to evenly spread employee group 1 (black dots) throughout the integrated list. Next, selected sections of employee group 2 were moved ("squeezed") incrementally toward the top of the list (right). The sum of these moves is shown in the final sample chart in Exhibit 17, which shows that the post-integration differential has been reduced for all employee groups, thereby achieving greater system equity. The results show that the majority of pilots are now able to retire with less than a 5-percent difference from their pre-merger retirement position. This is in contrast to an over 40-percent differential in some population segments before list adjustment edits.

While this editing example used retirement list percentage as the attribute of focus, others can also be modeled, such as final job level attained or compensation values. Further, proposal effects on attributes can be drilled down to specific time points (e.g., a specific month or year following integration) or slices of input data (e.g., all pilots hired in certain years, or who are within a certain age range). Various list conditions may be tested and compared as well. The final integrated list solution chosen would likely arise from a blending of attribute distortion reductions, confirmed by the other analysis capabilities of the program. ◼

Center for Hospitality Research
Publication Index
chr.cornell.edu

2017 Reports

Vol. 17 No. 11 The Billboard Effect: Still Alive and Well, by Chris K. Anderson, Ph.D.

Vol 17. No. 10 Ethics from the Bottom Up, by Judi Brownell, Ph.D.

Vol. 17. No. 9 Entrepreneurship Is Global: Highlights from the 2016 Global Entrepreneurship Roundtable, by Mona Anita K. Olsen, Ph.D.

Vol. 17. No. 8 Total Hotel Revenue Management: A Strategic Profit Perspective, by Breffni M. Noone, Ph.D., Cathy A. Enz, Ph.D., and Jessie Glassmire

Vol. 17 No. 7 2017 CHR Compendium

Vol. 17 No. 6 Do Property Characteristics or Cash Flow Drive Hotel Real Estate Value? The Answer Is Yes, by Crocker Liu, Ph.D., and Jack Corgel, Ph.D.

Vol. 17 No. 5 Strategic Management Practices Help Hospitals Get the Most from Volunteers, by Sean Rogers, Ph.D.

Vol. 17 No. 4 What Matters Most to Your Guests: An Exploratory Study of Online Reviews, by Jie J. Zhang, Ph.D., and Rohit Verma, Ph.D.

Vol. 17 No. 3 Hotel Brand Standards: How to Pick the Right Amenities for Your Property, by Chekitan Dev, Ph.D., Rebecca Hamilton, Ph.D., and Roland Rust, Ph.D.

Vol. 17 No. 2 When Rules Are Made to Be Broken: The Case of Sexual Harassment Law, by David Sherwyn, J.D., Nicholas F. Menillo, J.D., and Zev J. Eigen, J.D.

Vol. 17 No. 1 The Future of Hotel Revenue Management, by Sheryl E. Kimes, Ph.D.

2017 CREF Cornell Hotel Indices

Vol. 6 No. 2 Cornell Hotel Indices: First Quarter 2017: Status Quo Maintained, by Crocker H. Liu, Ph.D., Adam D. Nowak, Ph.D., and Robert M. White, Jr.

Vol. 6 No. 1 Cornell Hotel Indices: Fourth Quarter 2016: Hotels Are Getting Costlier to Finance, by Crocker H. Liu, Ph.D., Adam D. Nowak, Ph.D., and Robert M. White, Jr.

2016 Reports

Vol. 16 No. 28 The Role of REIT Preferred and Common Stock in Diversified Portfolios, by Walter I. Boudry, Ph.D., Jan A. deRoos, Ph.D., and Andrey D. Ukhov, Ph.D.

Vol. 16 No. 27 Do You Look Like Me? How Bias Affects Affirmative Action in Hiring, by Ozias Moore, Ph.D., Alex M. Susskind, Ph.D., and Beth Livingston, Ph.D.

Vol. 16 No. 26 The Effect of Rise in Interest Rates on Hotel Capitalization Rates, by John B. Corgel, Ph.D.

Vol. 16 No. 25 High-Tech, High Touch: Highlights from the 2016 Entrepreneurship Roundtable, by Mona Anita K. Olsen, Ph.D.

Vol. 16 No. 24 Differential Evolution: A Tool for Global Optimization, by Andrey D. Ukhov, Ph.D.

Vol. 16 No. 23 Short-term Trading in Long-term Funds: Implications for Hospitality Financial Managers, by Pamela C. Moulton, Ph.D.

Vol. 16 No. 22 The Influence of Table Top Technology in Full-service Restaurants, by Alex M. Susskind, Ph.D., and Benjamin Curry, Ph.D.

Vol. 16 No. 21 FRESH: A Food-service Sustainability Rating for Hospitality Sector Events, by Sanaa I. Pirani, Ph.D., Hassan A. Arafat, Ph.D., and Gary M. Thompson, Ph.D.

Vol. 16 No. 20 Instructions for the Early Bird & Night Owl Evaluation Tool (EBNOET) v2015, by Gary M. Thompson, Ph.D.

Vol. 16 No. 19 Experimental Evidence that Retaliation Claims Are Unlike Other Employment Discrimination Claims, by David Sherwyn, J.D., and Zev J. Eigen, J.D.

Vol. 16 No. 18 CIHLER Roundtable: Dealing with Shifting Labor Employment Sands, by David Sherwyn, J.D.

Vol. 16 No. 17 Highlights from the 2016 Sustainable and Social Entrepreneurship Enterprises Roundtable, by Jeanne Varney

Vol. 16 No. 16 Hotel Sustainability Benchmarking Index 2016: Energy, Water, and Carbon, by Eric Ricaurte

Vol. 16 No. 15 Hotel Profit Implications from Rising Wages and Inflation in the U.S., by Jack Corgel, Ph.D.

Vol. 16 No. 14 The Business Case for (and Against) Restaurant Tipping, by Michael Lynn, Ph.D.

CHR Advisory Board