Process Mining the Legislative Actions of the US Congress

Group 4 - 22/12/2021

Irene Fernández Rebollo, Mateo Jácome, Eliya Tiram, Kathryn Weissman

Professor Josep Carmona

Abstract. Process mining has become crucial in today's world, data can be found in mass amounts and it is rapidly growing, the necessity of process discovery is essential. In this project we will obtain real world data to create an event log, prepare it for analysis and perform an analysis to the presented process model. The project focused on the legislative actions in the US congress. The process, in this paper, performs a data extraction of different bills, the actions for each bill and when is taking place. We point out what was missing in the extracted raw csv file from the extraction process, and the steps performed to prepare it for the analysis stage. Lastly, we present an analysis of the process model, in Disco, including insights and statistics about the different bills

GitHub repository: https://github.com/KatBCN/PODS-Project

1. Introduction

As a part of the Process Oriented Data Science course project the legislation process in the United States has been explored and the data sources were scrapped. The main focus was on the 115th, 116th and 117th congresses and data was extracted from open source zip files that are open to the public. The congress is composed of two chambers, the Senate and House of Representatives. Since some congressional seats are up for election every two years, each individual congress consists of a set of members.

A bill, which is an idea for legislation, a form of a law before it passes, can be initiated by a legislator, which can be a member of the Senate, the House of Representatives or even the president. After a bill is introduced in a chamber, it is assigned to a committee. A bill can either pass or die in a committee, and in case it passes, it is voted upon in the respective chamber. If the bill passes one chamber, it is sent to the other for consideration. The bill repeats the process in the second chamber with its committees. If the bill passes both the House of Representatives and the Senate in any order, then it is sent to the president for consideration in order to become a law. The president can either sign the bill or veto it. If the president doesn't take action, the bill becomes a law automatically within 10 working days if congress does not adjourn.[1]

For each congress there are number of bill types which represents: House Bill (hr), Senate Bill (s), House Joint Resolution (hjres), Senate Joint Resolution (sjres), House Concurrent Resolution (hconres), Senate Concurrent Resolution (sconres), House Simple Resolution (hres) and Senate Simple Resolution (sres).[2] XML files are available on govinfo.gov as bulk data for each bill organized by bill type. The different XML files are zipped into one file per congress per legislation type.[3] Our goal is to scrape the data, clean it, and create an event log file that can be analysed with a process mining tool to discover a process model.

2. Data treatment

To begin with, a script in python was created to scrape the data, clean it and fill it when needed, with the purpose of preparing an event log file to analyse. First, the data were obtained from the different congresses files to get the very raw records from the different bills. After, data manipulation was performed to remove duplicates and fill missing fields that would allow for analysis with a process mining tool.

2.1. Data extraction and raw data

Data source

The data of the different bills is stored in govinfo. This website and repository is a service of the United States Government Publishing Office (GPO), which is a Federal agency in the legislative branch, to provide free public access to official publications from all three branches of the Federal Government.

The Bulk Data Repository offers different XML content for download, for the purpose of the project the Congressional Bill Status content is required where the information of the different bills is stored for different congresses since 113th Congress. Is possible to download a XML file by bill or a ZIP file with all bills in XML format for each congress and type. The types of bills, as described before, are hr, s, hjres, sjres, hconres, sconres, hres and sres.

In order to obtain an event log that represents this process model, only data from the three last congresses (115th, 116th and 117th) and from the four main bill types (hr, s, hjres and sjres) is downloaded to reduce the space occupied and make the problem feasible for a usual computer; the space needed to store this data is approximately 135MB with the files in ZIP format.

XML Files

In an XML file, there are both tags and text. The tags provide the structure to the data and the text with the information is surrounded by these tags. In the Bill Status files, there is a lot of information about each bill, but to obtain the event log, it is only necessary to identify the details of the bill and the different actions. These tags are:

- billNumber: e.g. *"2881"*
- billTitle: e.g. *"Secure 5G and Beyond Act of 2020"*
- billOriginalTitle: e.g. *" "*
- billType: e.g. *"HR"*
- congress: e.g. *"116"*
- actionDate: e.g. *"2020-01-08"*
- actionTime: e.g. *"16:07:41"*
- actionCode: e.g. *"H30300"*
- actionName: e.g. *"Motion to suspend rules and pass bill"*
- type: e.g. *"Floor"*
- sourceSystem/name: e.g. *"House floor actions"*
- text: e.g. *"Mr. Doyle, Michael F. moved to suspend the rules and pass the bill, as amended."*

Event log

In order to obtain an event log, the ZIP files, containing the XML files for each bill, are read in python to create a table with the tags named before as column names. Then, for each bill and action, the corresponding information is added to the table.
Once this first table is generated, some correction must be done such as removing duplicates and inserting actionCodes and actionNames missing or not correctly assigned.

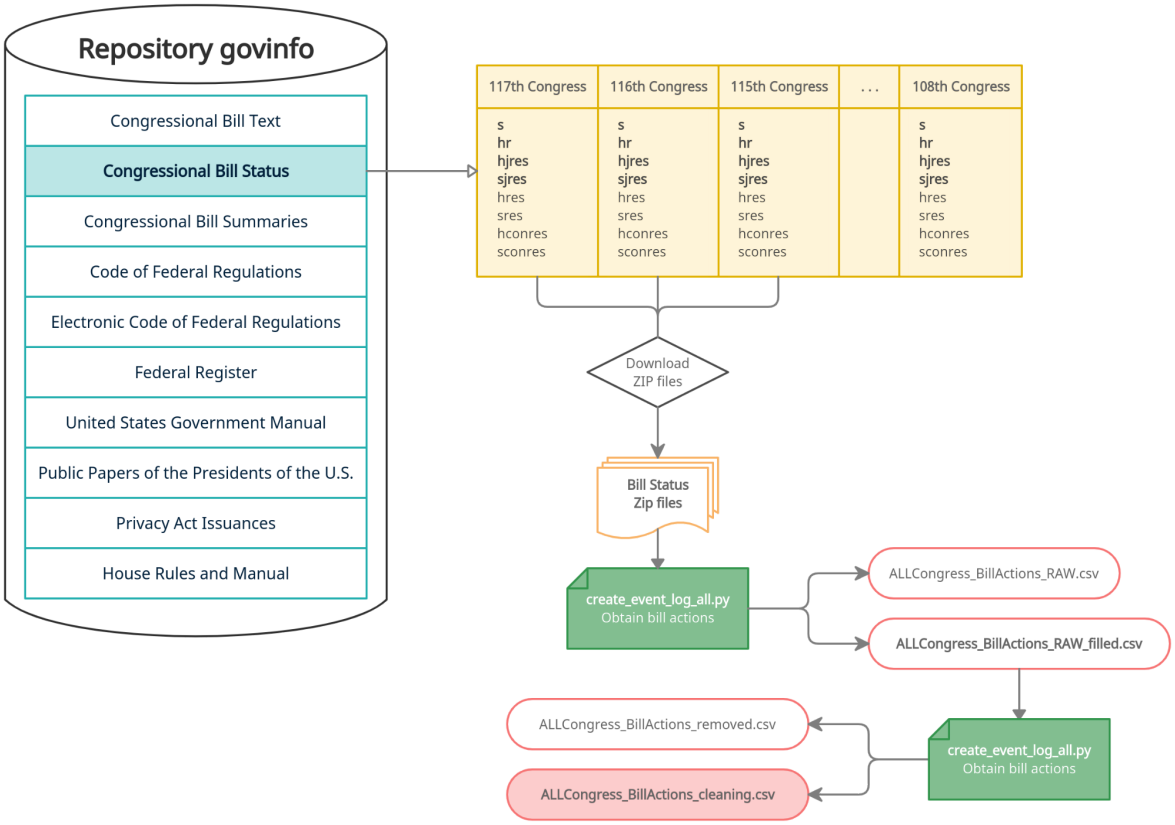


Figure 1. Diagram of data collection.

Action Codes

All actions taken from the bulk files contain an action type, and some contain action codes from their source systems. In the analysis of the 115th, 116th, and 117th Congresses, there are ten unique action types, and the three most common are “IntroReferral”, “Committee”, and “Floor.” 72% of all actions imported from the files have action codes provided. The code is typically a 4 or 5 digit number that sometimes has a letter at the beginning or the end of the digits.
Each source has a different level of granularity for recording actions. All of the actions from the sources “House floor actions” and “Senate” were missing codes. All of the actions from the source “House committee action” were of the type “Committee” which is the only action type used by all four source systems. There were 82 unique codes from the “House Floor Actions” and 30 unique codes from the “Library of Congress” source.

In order to interpret the codes in a meaningful way, one can reference the Library of Congress list of action codes.[4] Another useful reference is the “Bill Status XML Bulk Data User Guide” maintained by the GPO on GitHub.[5] Using these two references, we created the file `actionCode_dict.csv` that mapped 93 codes to actions in a field we created called “actionName” in the event log. The raw data contained codes that are not present in the mappings from either reference, which resulted in many missing values for the action name in the first version of the event log for the 117th congress.

115th, 116th, and 117th Congress Bulk Data	
Action type	**Number of activities missing a code**
Committee	25,312
IntroReferral	14,219
Floor	4,163
Calendars	1,627
Discharge	470
ResolvingDifferences	144
Veto	24

2.2. Data cleaning

Our approach to data cleaning was to start by using the 117th Congress bulk data as a working set. Once satisfactory results were achieved for the cleanliness and ability to perform analysis, the dataset was expanded by incorporating bulk data from the 115th and 116th Congresses. Expanding the dataset was an important step, because the 117th Congress is only half-way through it's term at the time of our analysis.

New Action Code Assignment

In order to assign codes to the actions that didn't have one, a set of rules was developed that depended on the source system, the action text, and the code mappings published online by the Library of Congress and the GPO. Almost all of these rules were created using the 117th Congress dataset, and then a few more rules were added related to vetoes once the dataset was expanded.

Developing the rules was not an automatic process, and it involved reading the action text and searching for a suitable code from the published set of codes. Google Colab notebooks were used to explore and analyze the data.[6] When a suitable code was not found for Senate data, a new code was created attempting to follow the same level of granularity used by the House of Representatives. The rules are implemented in the function `fillCode(row)` of `create_eventlog_all.py`. In total, 26 rules were created to assign codes based on specific substrings appearing in the action text field. After passing the rows of data with null values to the function, the resulting dataset was not missing any action codes. Below is an example of the implementation.

> Text from HR 116 "Banking Transparency for Sanctioned Persons Act of 2019": "Received in the Senate and Read twice and referred to the Committee on Banking, Housing, and Urban Affairs."

The first rule relating to the Senate:

```
elif row['sourceSystem/name'] == "Senate":
    if "received in the senate" in str.lower(row['text']):
      return '10000'  # Introduced in Senate
    elif "introduced in the senate" in
str.lower(row['text']):
      return '10000'  # Introduced in Senate
```

The above example highlights some of the difficulty of automatically processing action text to fill missing data. The action text related to the bill actually describes two actions, an introduction to the senate and also a referral to a committee. A committee referral has its own code if it is the only action present in a row. There is an interest in analysing when bills are passed from one chamber of the legislature to the other, so the code relating to introduction was deemed more important, which is why it is the first rule in the list of if-else statements in the function.

Other cases are even more complex, and there are likely errors in the event log due to a misunderstanding of procedural terminology or the importance of an action. It is possible, however time consuming, to do spot checks for completeness of a bill's action codes by using Congress.gov's advanced search tool as described by librarian Beth Osbourne in a law library blog post.[7][8] The quality check is not guaranteed to work as intended. For example, our data contains the action code 41000 "Became Private Law" two times for the same law, but there is no record of this law in the advanced search tool. It is unknown how coordinated the legislature's database systems are with the public bulk data and various online tools.

New Action Name Assignment

Due to action codes being present in the bulk data which were not included in the set of reference codes, new action names had to be assigned to codes. A similar approach was followed for assigning action names as was used for assigning action codes. The text was read, and the references were analyzed to check if the code could be considered duplicative. Codes associated with similar texts were assigned identical action names, with the intention of using the name instead of the code as the event field in tools like Disco.[9]

The output of action code and name analysis was an updated version of the code mappings in the file `updatedCodes_dict.csv` which was used for filling action codes and names to create the final version of the event log.[10] 53 mappings were added to the original set of 93.

Data Issues Handling

When the data was obtained for the very first time a number of issues were raised: missing time value to some of the records, duplicate records, redundant records - different action codes with the same meaning on the same day. The next part describes how the issues above were solved and handled.

In order to solve missing action time values, the time `00:00:00` was assigned for each action that doesn't have action time on the XML. By performing this action a full date was populated for all of the log records with the same format.
Since there are duplicate records in our log, the method applied on the main dataframe is `drop_duplicates()`. It simply returns a dataframe with no duplicate values for a whole record.

Another point to solve was redundant data, having records with different action code but the same meaning needed to be handled. When the raw csv file was ready, as part of the cleaning process, we looped over it and defined that a given bill title with the same action name on the same date cannot occur more than once. Within the loop the indexes of the redundant records were saved and were dropped later on from the dataframe.
All the deleted records were stored in a new csv file to audit the process in case it will be needed.

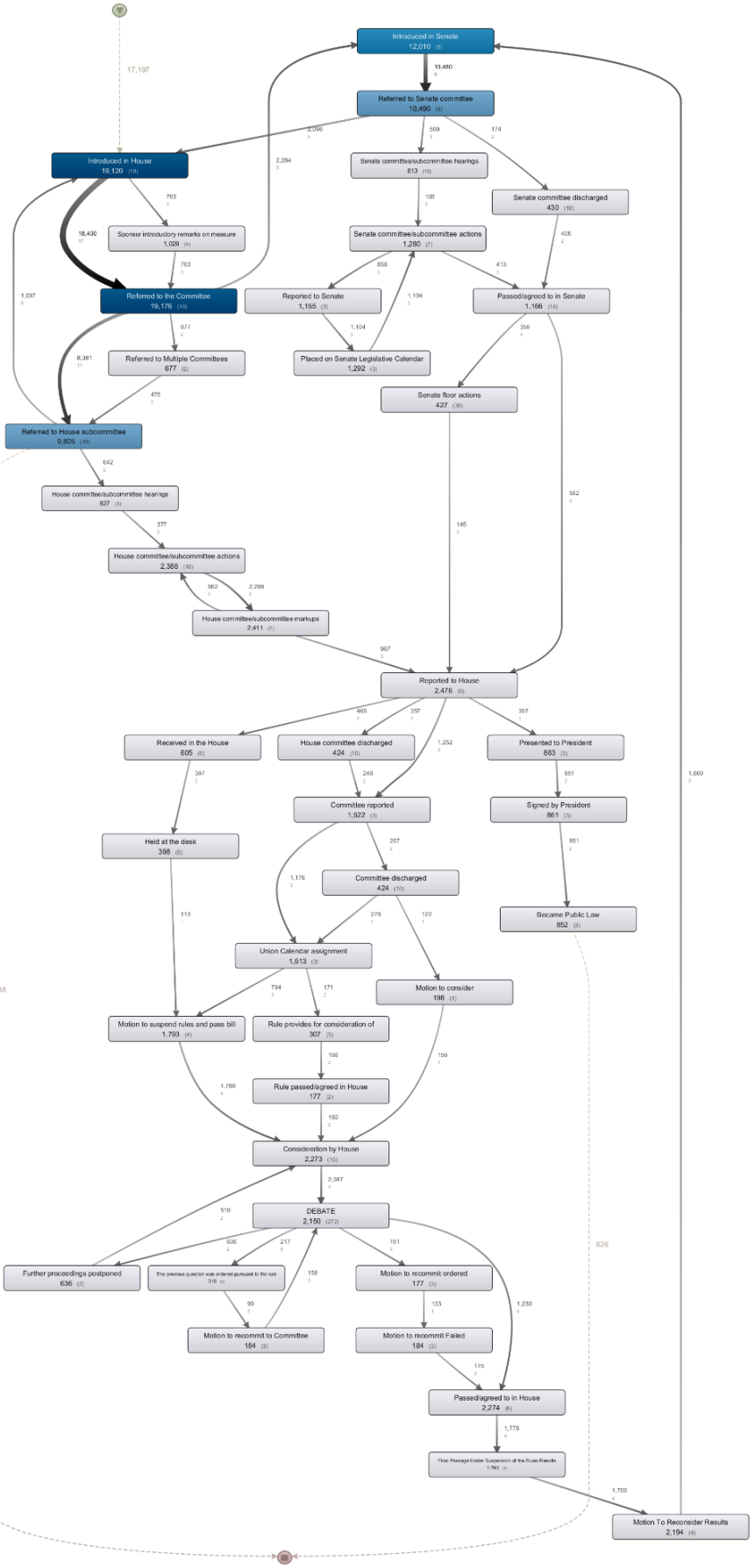
On the next step, the data was loaded to Disco for the first observation. It was seen that action names which can happen on the same day were not in the proper order. In order to solve this problem the part of the time within the full date was modified to be `23:59:59` for the latest action, `23:59:58` for the one action before the latest and so on.

3. Data Analysis

3.1. Analysis tool

We've decided to use Disco as a tool for process analysis given its robustness and simplicity. We loaded the clean event log into the software and selected `billTitle` as ID, `fullDate` as timestamp, `actionName` as activity. From the point of view of our process, each bill will be one case. In addition, one can set `congress`, `billType`, or `type` as 'other' to use them to get insights into specific data sources or congresses.

Below we are adding a figure with our process map based on Frequency, with max repetitions as a secondary measure. However, we recommend consulting it in our GitHub repository for higher resolution. In addition to this process map, we also produced another based on Performance with mean durations as primary measure and case frequency as the secondary. This map can also be consulted in our GitHub repo.


17,197
Introduced in Senate
12,010 (9)
10,480
Referred to Senate committee
10,490 (9)
2,086
509
174
Introduced in House
19,120 (18)
2,284
Senate committee/subcommittee hearings
813 (16)
Senate committee discharged
430 (16)
763
193
406
18,406
Sponsor introductory remarks on measure
1,009 (4)
Senate committee/subcommittee actions
1,280 (7)
763
858
413
1,037
Referred to the Committee
19,176 (18)
Reported to Senate
1,155 (3)
1,104
Passed/agreed to in Senate
1,166 (10)
677
1,104
358
8,381
Referred to Multiple Committees
677 (2)
Placed on Senate Legislative Calendar
1,292 (3)
475
Senate floor actions
427 (36)
Referred to House subcommittee
9,805 (19)
642
House committee/subcommittee hearings
827 (3)
362
377
145
House committee/subcommittee actions
2,386 (16)
962
2,298
House committee/subcommittee markups
2,411 (5)
967
Reported to House
2,476 (5)
465
257
367
Received in the House
605 (5)
House committee discharged
424 (10)
1,252
Presented to President
863 (3)
397
249
851
1,660
Committee reported
1,522 (3)
Signed by President
861 (3)
Held at the desk
398 (5)
257
851
1,176
Committee discharged
424 (10)
113
278
122
Became Public Law
852 (3)
Union Calendar assignment
1,513 (3)
7,138
794
171
Motion to consider
198 (1)
Motion to suspend rules and pass bill
1,793 (4)
Rule provides for consideration of
307 (3)
166
166
826
1,789
Rule passed/agreed in House
177 (2)
160
Consideration by House
2,273 (16)
2,087
519
DEBATE
2,150 (272)
936
217
151
Further proceedings postponed
636 (2)
The previous question was ordered pursuant to the rule
318 (8)
Motion to recommit ordered
177 (3)
158
99
133
1,230
Motion to recommit to Committee
184 (3)
Motion to recommit Failed
184 (2)
175
Passed/agreed to in House
2,274 (6)
1,778
Final Passage Under Suspension of the Rules Results
1,793 (4)
1,700
Motion To Reconsider Results
2,194 (4)

3.2. Process analysis

3.2.1. Process map

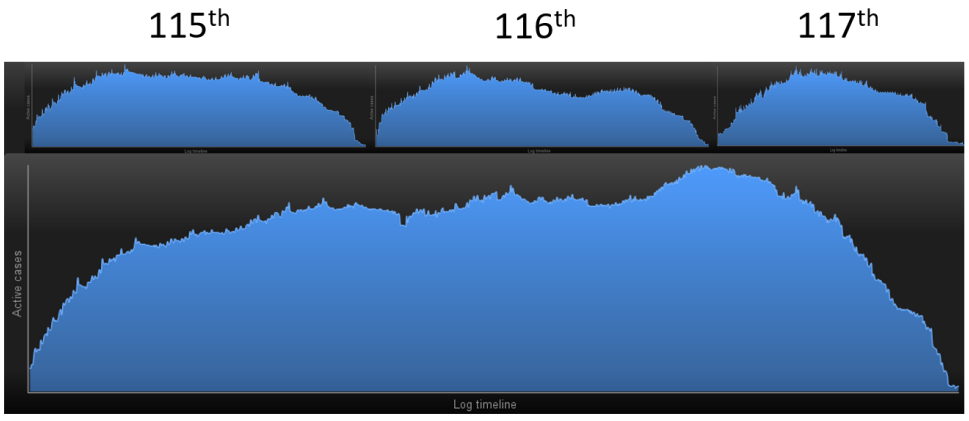
Our process map quickly reflects several points of interest about the legislative process of the US Congress. We can see how most of the bills are first introduced into the House of Representatives (although many are first introduced to the Senate, which the diagram does not show). We can also see how most of those bills only advance one or two steps, with only a small fraction of the cases actually going forward. We can also observe how the House and Senate work in parallel.

Some bills will only pass through one of the two chambers, while some will go through both Senate and House. We can see the points where the House's process and the Senate's process connect for those bills that pass through both. The bills that enter the Senate and are then referred to the House do so after being passed on by the Senate. The bills that enter the House and then are referred to the Senate are mostly going through the process of Debate, then passed by the house, and then referred to the Senate.

However, as one can see through the number of cases going from one action to another, this map is very incomplete, showing only the most common activities and paths for the sake of understandability. The process is in fact much more complex, with specific cases taking very intricate paths.

3.2.2. Workload over time

All Bills

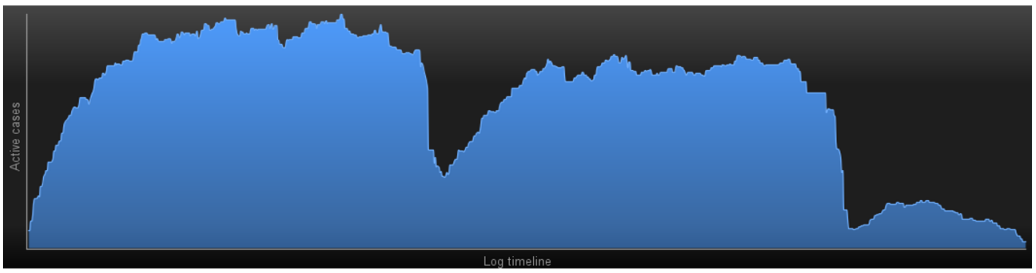


All congresses
All bills

Each of the congresses we analyzed shows the same pattern when studying active cases over time. The active cases grow rapidly and stabilize in a plateau towards the half of the first year of the 2 years for which each congress is instituted. Then, the number of active cases gradually drops as the number of registered actions decreases towards the end of the congresses, which causes cases to be considered inactive. However, when looking at the event log of the three congresses combined, it seems that the number of active cases doesn't drop significantly between different congressional periods, partially due to unfinished

bills being reintroduced in the next congresses. We can observe a surge in active cases before cases decline in the last quarter of the graph showing the active cases for all the congresses. This surge in active cases coincides with april 2020 and may be caused by a legislative response to the impact of the COVID-19 pandemic.

Bills that became law

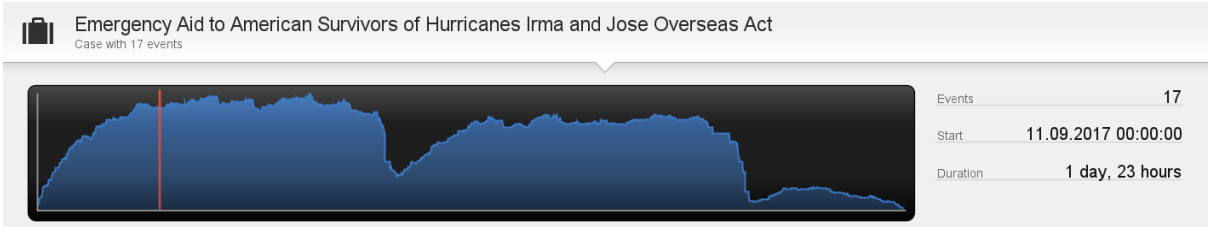


All congresses
Passed bills

When studying only the complete cases, meaning bills that have passed and became law, we can observe a different pattern, as seen in the figure below. At the end of every Congress, all active cases are dropped, and any bills that the members of the next Congress still want to pass need to be reintroduced. This, combined with the fact that approximately 35% of the bills that become law during a Congress do so in the last three months of the same [11], causes a sharp fall in the number of active cases at the end of each Congress. As one could expect given the previous facts, because the 117th congress is still ongoing, the number of bills that became law until this point is very low, which is reflected in the graph.

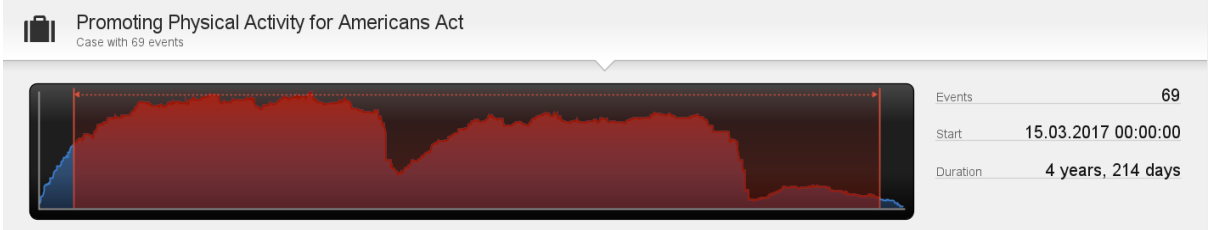
3.2.3. Case durations

Bills can pass very quickly, such as the *"Emergency Aid to American Survivors of Hurricanes Irma and Jose Overseas Act"*, which was introduced in the House of Representatives one day and became law the next. Other natural catastrophes and devastating events, including other hurricanes or the COVID-19 pandemic have also caused bills to be introduced, passed, and signed by the president in very short periods of time, which comes to show how the process we're describing can be very flexible.



However, the process can be much slower, such as in the case of the *"Promoting Physical Activity for Americans Act"*, which taking only the 115th to 117th congresses was the slowest bill to be passed and turned into law. A total of 4 years and 214 days passed since it was

introduced in 2017 in the Senate during the 115th Congress until it became law during the 117th Congress in 2021. This makes it the slowest bill to pass in the period we're observing.



3.2.4. Process bottlenecks

In the congressional road to becoming a law, most bills die out in the first steps. Out of the 24,805 cases in our event log, ~9,000 cases (38%) were only introduced into one of the two chambers and then referred to a committee, not advancing any further. ~3,700 cases (15%) were introduced, then referred to a committee, and lastly referred to a subcommittee. This points towards the main bottleneck in the process, where more than half of the bills die: the committees.

Some of the bills that are effectively forgotten in the committees during a congress can be reintroduced in the following ones, although this doesn't happen often. In the case of the House of Representatives in our observed timeline, of the ~19,000 cases making it to a committee, ~9000 make it to a subcommittee, ~7,000 end there, and ~2,800 are reintroduced into the House later in time. Of the bills passing to a subcommittee, ~7,100 end there, while ~600 make it to hearings and achieve further progress, and lastly ~1,100 bills are reintroduced in the House at a later point in time. Both steps show a very low/mild success rate (12-45%), and a low high failure rate (47%-78%). Similar proportions can be obtained when analyzing the Senate.

Overall, of the 24,805 bills only 826 eventually became law. Again, we must take into account that some of these bills belong to an unfinished Congress and can still progress without being reintroduced. By looking only at the finished congresses, we can find a success ratio of 4,6%.

3.2.5. Common variants

While we can see how most bills are left out in the first steps by looking at the most common variants for all cases, when we look at the common variants for bills that eventually become law we can find some curious observations. Given the complexity of the process, one would expect that all laws pass leaving very unique traces. However, some types of bills are processed following a very uniform process, and sometimes they are even processed in batches.

Interestingly, the 7 most common variants are composed of 7 to 23 cases each. Each of those repetitive cases are bills passed to change the name of a United States Postal Service office to honor a veteran, a remarkable citizen, or a public figure. One of them named after Marylyn Monroe, and another after Ritchie Valens, composer of the ageless hit 'La Bamba'. Given their superficial importance and small controversy, these bills are processed in a very

streamlined manner, causing the apparition of these common variants, and often pass quickly.

On other occasions, we can find small groups of more complex bills that share the same traces. Those are often related in topic are partially processed in the same sessions, such as the *"DEBAR Act of 2021"* and the *"Ensuring Compliance Against Drug Diversion Act of 2021"*, both of which were introduced and passed in different points in time, but were debated and partially processed jointly in the same House of Representative sessions.

4. Conclusions

Opportunities for Improvement

As the team worked on this project, we identified areas for improvement and plans for next steps if there were more time and resources available.

Data Processing

Implementing some advanced text mining techniques during the cleaning of action codes and names could reduce or eliminate the need for manual interpretation of activity text. The difficulty of automating that process is that the four source systems providing the text are not necessarily coordinated or consistent. Our process could also become more dynamic. Our script reads a set of downloaded files and creates an event log from scratch each time. Since the government updates the XML files on a daily basis, it could be useful to maintain a master event log and update only cases which have changes.

Adding Resources

A future improvement to the event log could be adding resources so that bottlenecks could be identified based on which person or committee is performing certain activities. It is common for bills to die in committee, so it could be interesting to see which committees are more successful at advancing bills. There are 435 representatives and 100 senators, so it could be time-consuming to ensure each action is properly identified, but adding legislative members would allow analysis of strongly connected component graphs to understand relationships among members.

Adding resources to the log would increase the number of activities being tracked, since bills can be referred to many committees at the same time. For example, our data processing will only record "Referred to House committee" once per day per bill, but often a bill is referred to many committees on the same day.

Adding Actions

There are other activities with dates documented in the raw XML files that could be imported to the event log. Adding more activities of different types will make the data preparation process more complex because activities are recorded in different levels of detail, and some

may be duplicates of already recorded actions. The XML files have containers for amendments, cboCostEstimates, committees, cosponsors, and recordedVotes which were considered to be too detailed or duplicative of other actions we decided to import.

Conformance Checking

The legislative process has been established for many years, changes infrequently, and is well documented and researched by many stakeholders. There are articles and resources available online that document the legal process which can be used to check the validity of our process model and the quality of our event log. A large, complex diagram of US Congress legislative status steps was published in 1974 and is still referenced by the GPO.[12] One of the next steps in this project could be conformance checking.

References

[1] “The Legislative Process”, *house.gov*, U.S. House of Representatives, accessed 22 December 2021, https://www.house.gov/the-house-explained/the-legislative-process
[2] “Congressional Bills”, *govinfo*, U.S. Government Publishing Office, updated 18 March 2021, https://www.govinfo.gov/help/bills
[3] “Bulk Data”, *govinfo*, U.S. Government Publishing Office, accessed 22 December 2021 https://www.govinfo.gov/bulkdata/BILLSTATUS/
[4] “Field Values for Legislative Action Codes.” Congress.gov, Library of Congress, accessed 21 December 2021, https://www.congress.gov/help/field-values/action-codes.
[5] “Bill Status XML Bulk Data User Guide”, GitHub, United States Government Publishing Office, updated 7 December 2020, https://github.com/usgpo/bill-status/blob/master/BILLSTATUS-XML_User_User-Guide.md#3-action-code-element-possible-values
[6] MissingActionCodes.ipynb, GitHub, KatBCN/PODS-Project, https://github.com/KatBCN/PODS-Project/blob/44c31543f07a7f3601fd662f069d968f8b5fc7d0/MissingActionCodes.ipynb
[7] “Advanced Search - Legislation”, Congress.gov, Library of Congress, accessed 21 December 2021, https://www.congress.gov/advanced-search/legislation
[8] Price, Anna & Osbourne, Beth. “Advanced Searches Using Legislative Action Codes on Congress.gov”, Law Library Blog, Library of Congress, 7 January 2021, https://blogs.loc.gov/law/2021/01/advanced-searches-using-legislative-action-codes-on-congress-gov/
[9] Disco, Fluxicon, accessed 22 December 2021, https://fluxicon.com/disco/
[10] updatedCodes_dict.csv, GitHub, KatBCN/PODS-Project https://github.com/KatBCN/PODS-Project/blob/44c31543f07a7f3601fd662f069d968f8b5fc7d0/obtain_data/updatedCodes_dict.csv
[11] “Statistics and Historical Comparison”, *govtrack*, Civic Impulse LLC, accessed 22 December 2021. https://www.govtrack.us/congress/bills/statistics
[12] “U.S. Congress legislative status steps“, Prints & Photographs Online Catalog, Library of Congress, accessed 22 December 2021, https://www.loc.gov/pictures/item/2012648836/

Appendix

Teamwork Distribution

All four team members met each week to discuss roadblocks and make decisions about how to move forward to achieve the best event log possible.

Research about the topic and available data sources and resources was led by Kathryn.

Writing python scripts was led by Irene and Mateo with support from Kathryn and Eliya.

Analysis of action codes and action names was done jointly by Irene and Kathryn.

The action code dictionary was created by Mateo and updated by Kathryn.

Disco analysis and modifying the data cleaning routine was led by Mateo.

Report writing was led by Eliya with support from all team members.

The following team members authored each section of the report:

- Abstract - Eliya
- Introduction - Eliya
- Data Treatment
    - Data Extraction and raw data
        - Data Source - Irene
        - XML file - Irene
        - Event log - Irene
        - Diagram of data collection - Irene
        - Action Codes - Kathryn
    - Data Cleaning
        - New Action Code Assignment - Kathryn
        - New Action Name Assignment - Kathryn
    - Data issues handling - Eliya
- Process Analysis - Mateo
- Conclusions
    - Opportunities for Improvement - Kathryn
- References - Kathryn
- Summary Statistics of Event Logs - Kathryn

Individual Report: Irene Fernández Rebollo

The course of Process-Oriented Data Science was very useful to understand how to manage and analyze the processes of any type of organization. After this time, I am much more interested in this field and I have been able to comprehend the relevance of the processes nowadays.
I work as a bioinformatician and I thought that this subject was not related, but I figured out that we can extract some processes from anywhere and how important it is to improve any task.

The project was challenging, in our case, we tried to analyze the process of a bill in the United States to become a law. We choose this data because we want to analyze a process that has not been analyzed and publicly documented, this has the problem that the data is not completely cleaned and there is not any reference of how to manage it.
The most interesting part for me was how to download the proper data and finally generate an event log. This is the task in which I work more. I used python scripts to obtain the required information of the XML files and transform it in tables to create the event log. These are tasks, I have not done them before, but it was very interesting how to transform the data structure and the different ways to clean the data, improving the quality.
There is still a lot of work that could be done, go in depth in the analysis of the process by using other tools and try to separate clearly the different sources or get some more metrics about the actual diagram, but I think that we worked hard and we obtain a good result where we can get some insights of the legislative actions process.

Individual Report: Mateo Jácome

On the course

I've found PODS to be very interesting for multiple reasons and scopes. First of all, process mining and process analysis comprise an incredibly applicable and powerful set of techniques from which any organization could take advantage. I feel like the majority of what we've learnt could have direct use in a professional setting. The different algorithms and software we've peaked at are capable of reconstructing processes from scratch with ease, as long as they're fed a good event log.

They are in addition very powerful and have a great capacity for giving insights into numerous fields, such as studying process performance and conformance: finding weak spots, bottlenecks, and incongruencies becomes surprisingly easy. In fact, the sharpness and strength of these tools leave me with both a preoccupation and a brim of hope. A preoccupation because it's easy to understand how these tools can form a power couple with bad labour policies in companies. While a company's process can surely be improved in many ethically correct ways, some aspects of it can become troublesome, such as bringing the efficiency and ease of use of process mining and analysis together with automation of human resources departments. For example, it would be easy to pressure employees based on their performance without properly checking on them as workers, and well, humans.

However, at the same time, I see a clear brim of hope. The very same techniques that could be exploited for ill purposes can be used for the very opposite: auditing good practices in companies, validation of standards such as ISO norms, checking that laws are properly respected and implemented in companies and institutions… and a very long *et cetera*. The uses of these techniques are very versatile, and I suppose it will be in our hands to use them properly and fairly in our future careers.

In a different tone, and as a biotechnologist, I'm very curious about the potential future implementations of process mining and cellular biology. As the field of synthetic biology quickly advances, new techniques that enable external recording of cellular processes could enable the logging and creation of biological event logs which would be of incredible interest to study the behaviour of both individual cells and simple cellular systems such as organoids or early developing embryos. Given the intricate complexity of living systems, process mining and analysis could yield very interesting results. Actually, it's something I've often hesitated to ask about in class given that the technology to properly implement PODS may not be fully available now, but it will doubtlessly be in the upcoming years. In any case, I believe at least for now it's an interesting thought experiment: exploring today which biological systems could be analysed using the techniques we've studied, and which biotechnological tools are available for it today or would need to be developed could have a great potential in the future of synthetic biology.

On the project

When Kathryn first proposed the project, I thought it could be really cool and a bit out-of-the-box. Then I saw the data that we would be using as a source for the first time, and I was mortified. I had never worked with XML files before, and I wasn't sure how we'd be able to deal with something like that in order to produce the event log. However, all of us got our hands dirty and quickly came up with solutions which slowly tackled each of the problems we faced during the first steps. First, parsing what information from the XML bulk data was of use for our goals and extracting it. Then, coming up with a way to combine the data from one source (such as the Senate, or the House of Representatives) and the others for a given congress, and lastly, obtaining and combining the data from different congresses.

All of these main challenges required a lot of research and effort to solve, and often yielded smaller sub-challenges that needed to be solved such as assigning codes and names to actions that were missing those fields. Once we had an event log we were happy with, it was very fun to explore it and try to come up with insights on the process that were interesting. Overall, I would say that we've done a quite good job, since our process map clearly reproduces a good part of the legislative process carried out by the U.S. congress. I believe we've got to a very good point from which one can already analyse the process, although much more could be done in different directions. A very interesting point would be incorporating information about the people and systems who actually compose each of the congress chambers – many of the actions we've recorded and analysed are explicitly performed by one congressperson or committee, and that would be a very rich source of information for more in-depth analysis. Another interesting point would be using the process we have studied as an example of conformance. By trying to achieve a wholly complete event log where every activity registered by the Congress' institutions is included, one could try to study conformance to the established rules. Given the importance of such a process, the amount of resources and people (both internal and external to the Congress) dedicated to checking every step that is made, and the rigor with which it's implemented and checked, it may very probably be an almost perfect process in terms of compliance to the rules.

To conclude, I'm glad to say that I'm very pleased with how we all have worked individually and as a team. I feel like we all have perceived this project as a very interesting one, both for the topic, the tools, and for the challenges we have had to solve, and that has turned out in a lot of good effort put on, a great grade of implication, and very good teamwork.

Individual Report: Eliya Tiram

I have some familiarity with data processes and the course was giving me a new point of view of process mining. When I think about my career and the way I want it to develop I find that what we have learnt in the course, and in this project, will be helpful as it gives me the opportunity to also have a practical experience on process mining. As I see it, data mining can have a variety of ways whether it's the way it is implemented, the technology it uses, or the source of the data (database, csv file, xml file, etc.). I think that the process map of a log or a business process obtained from an organizational data can benefit both the people within an organization and the organization itself.

Working on the project was challenging and scholarly, I had a chance to see how to work with XML, which is different from sources I was working with in the past. Working with a group was fun and motivated. For me personally, it is the largest group I was working with in UPC and it is nice to have their support and being able to raise questions in our group.
I felt comfortable that we used python scripts as I think it's the proper code for this kind of task, it is readable and for me easier to understand than other coding languages. In the project I think it was challenging to get data which is missing but deep within have a meaning that needs to be found. This part was new to me because I used to work with a data supplier which took liability on the data in case it had missings and here we needed to align it.
To sum up, I really related to the project and enjoyed working on it as part of the group and individually, I learn a lot from the process and I really feel like I want to keep our code and process as it might be handy in the future.

Individual Report: Kathryn Weissman

From the beginning of the course, I could relate to the topic based on my previous professional experience working as a project manager for a US government agency. Process mining using event logs of IT systems seems to be a very useful tool for tracking conformance of regulated processes. As part of my job, I managed competitive procurement of new contracts and was also involved in invoice processing.

Any task which involved financial accounting was processed through IT systems using individual user accounts. I never saw event logs from the systems, but now I'm sure they could have been used to check conformance with regulations or to manage KPIs. We often had KPIs about the timeliness for processing invoices or contract modifications. The acquisitions process for new contractors was highly regulated, and every step of the competitive procurement process was managed through multiple online systems with actions flowing from one user to the next with checkpoints for approvals by users with authority.

I think one of the big challenges for process mining will be the quality and accuracy of real event logs. From my personal experience, I know that when users are aware of how timeliness KPIs are calculated, they modify their behavior to do coordinated work outside of the official systems, and wait until everything is ready to start processing a case. This would especially happen during holiday periods when the office was not fully staffed, as it was known that it would take longer to process cases. The person initiating the case would be asked to wait until certain people were available. This makes the workflow appear to be faster than it actually is. We should be cautious to rely on event logs as "the truth."

By working on the project, and reflecting about my own experience using systems, I see that it can be very difficult to extract a high quality event log. Many users will not fill in optional data which can make it difficult to compare activities. The project would have been a lot easier if each source system of the legislature recorded activities at the same granularity and the activities were well coded. A system that is easy for the user entering data is not necessarily easy for a data analyst. The congressional systems often recorded more than one activity in a single action which also makes it difficult to ensure that no steps are missed in the process.

At the beginning of the project, the first challenge was finding suitable open source data that could be used to analyze an event log. Kaggle is used in other courses because it has so many datasets, but it was very difficult to find interesting data that could be turned into an event log. Once I discovered the legislative process data, then the challenge was limiting the project to a reasonable scope that could be completed in a short period of time by students. I think that there will be opportunities available for data scientists at the Library of Congress and other public institutions to document and research public processes.

During the project, I learned about XML files for the first time, and became more familiar with GitHub and Google Colab Notebooks. I liked using the Colab Notebooks for exploring the event log and analyzing the action codes and action names. It was convenient to import the event log as a pandas dataframe and then use different functions and methods to sort and filter the data quickly. Since filling missing action names was a manual process, I used the notebooks to iteratively view all texts with the same action code and look for similarities by

skimming the list. In the future, I will try to learn more about how to apply text mining algorithms to make it a more automatic process.

Working as a team was very helpful for this project, because each person had different skills and knowledge to contribute which made the tasks manageable. Whenever some of us had doubts about our ability to proceed in a certain way, another person would offer encouragement, an idea for a solution, or take initiative for the next step. We ended up analyzing an event log with more than 24,000 cases which I did not think was possible at the beginning of the project.

Data Processing: Summary Statistics of Event Logs

The statistics are produced using the file `CheckingEventLogs.py`

Comparing Event Log files updated on 22 December 2021:

- `ALLCongress_BillActions_RAW.csv`
- `ALLCongress_Bill_Actions_RAW_filled.csv`
- `ALLCongress_Bill_Actions_removed.csv`
- `ALLCongress_Bill_Actions_cleaning.csv`

Summary Statistics of Event Logs during Data Processing Steps				
	Raw Bill Actions	**Complete Codes & Names**	**Removed Actions**	**Final Event Log**
Unique Bill Titles	24,805	24,805	19,240	24,805
Rows	166,790	166,790	25,367	141,423
Unique Action Codes	110	123	5	121
Unique Action Names	71	82	3	82

Number of Actions from each Source System				
	Raw Bill Actions	**Complete Codes & Names**	**Removed Actions**	**Final Event Log**
Library of Congress	71,061	71,061	23,998	48,401
House Floor Actions	49,770	49,770	1,369	47,063
Senate	24,132	24,132	0	24,132
House Committee Actions	21,827	21,827	0	21,827

Number of Actions of each Bill Type				
	Raw Bill Actions	**Complete Codes & Names**	**Removed Actions**	**Final Event Log**
HR	125,771	125,771	24,070	101,701
S	37,855	37,855	839	37,016
HJRES	1,988	1,988	410	1,578
SJRES	1,176	1,176	48	1128

Percent of Actions in Dataset by Type				
	Raw Bill Actions	**Complete Codes & Names**	**Removed Actions**	**Final Event Log**
IntroReferral	58.25%	58.25%	89.86%	52.58%
Committee	18.56%	18.56%	-	21.89%
Floor	17.46%	17.46%	3.41%	19.98%
Calendars	1.94%	1.94%	-	2.28%
President	1.56%	1.56%	3.37%	1.23%
Discharge	.65%	.65%	-	.76%
Became Law	1.02%	1.02%	3.36%	.61%
ResolvingDifferences	0.44%	0.44%	-	.51%
NotUsed	.11%	.11%	-	.13%
Veto	.02%	.02%	-	.03%

Console Output

The following statements are printed to the console while running the file `create_eventlog_all.py` that creates the event logs in order to track progress. The execution time varies, and generally lasts between 30 minutes and one hour to read the data from 3 congresses.

```
Starting process 2021-12-22 14:24:39.536056
116
Started STATUS-116-sjres.zip 2021-12-22 14:24:39.538147
Finished STATUS-116-sjres.zip 2021-12-22 14:24:40.088716 took
0:00:00.550574
Started STATUS-116-hr.zip 2021-12-22 14:24:40.149726
Finished STATUS-116-hr.zip 2021-12-22 14:28:14.412948 took
0:03:34.263232
Started STATUS-116-s.zip 2021-12-22 14:28:14.461480
Finished STATUS-116-s.zip 2021-12-22 14:32:37.729428 took
0:04:23.267960
Started STATUS-116-hjres.zip 2021-12-22 14:32:37.731857
Finished STATUS-116-hjres.zip 2021-12-22 14:32:44.416506 took
0:00:06.684658
Finished process 2021-12-22 14:32:44.416659 took 0:08:04.880604
117
Started STATUS-117-sjres.zip 2021-12-22 14:32:44.803921
Finished STATUS-117-sjres.zip 2021-12-22 14:32:45.878595 took
0:00:01.074682
Started STATUS-117-hjres.zip 2021-12-22 14:32:45.880459
Finished STATUS-117-hjres.zip 2021-12-22 14:32:48.157000 took
0:00:02.276547
Started STATUS-117-hr.zip 2021-12-22 14:32:48.180701
Finished STATUS-117-hr.zip 2021-12-22 14:45:51.664687 took
0:13:03.483993
Started STATUS-117-s.zip 2021-12-22 14:45:51.678957
Finished STATUS-117-s.zip 2021-12-22 14:49:57.371084 took
0:04:05.692147
Finished process 2021-12-22 14:49:57.371252 took 0:25:17.835197
115
Started STATUS-115-hjres.zip 2021-12-22 14:49:57.566502
Finished STATUS-115-hjres.zip 2021-12-22 14:50:05.470074 took
0:00:07.903580
Started STATUS-115-sjres.zip 2021-12-22 14:50:05.471529
Finished STATUS-115-sjres.zip 2021-12-22 14:50:09.250876 took
0:00:03.779354
Started STATUS-115-hr.zip 2021-12-22 14:50:09.279384
Finished STATUS-115-hr.zip 2021-12-22 15:13:27.349049 took
0:23:18.069672
Started STATUS-115-s.zip 2021-12-22 15:13:27.364872
```

```
Finished STATUS-115-s.zip 2021-12-22 15:20:27.131673 took
0:06:59.766809
Finished process 2021-12-22 15:20:27.131806 took 0:55:47.595751
(166790, 11)
45959 actions with null codes before calling fillCode()
0 actions with null codes after calling fillCode()

45959 actions with null names before calling fillCode()
45991 actions with null names after calling fillCode()
51 actions with null names after merging


Top 5 codes without name and their counts:
SenateMotionDischarge    19
H40210                    4
H40141                    4
H32500                    4
H40300                    3
Name: actionCode, dtype: int64
```