ValerieMJones

IST 652 | Scripting for data analysis

iSchool Syracuse University

pdf scraping for army national guard state and service award identification and eligibilty

1. **Introduction**

As a 20 year member of the Army National Guard and the Current First Sergeant for Maverick Company, Train Battalion[[1]](#footnote-1), I have witnessed the evolution of the technological changes to Personnel Information Management. Unfortunately, the National is still behind the curve in automation and still rely on manual updates of data transactions that impact a Soldier’s career. Over the last ten years I have worked to leverage my Data Analysis background to improve internal processes within my assigned sections and Company.

1. **Initial Project Scope**

Each month Operations Company, Headquarters and Headquarters Battalion, 28th Infantry Division is responsible for an assigned detail. There are four main detail areas each rotated through by all four companies of the battalion each month. As the Operations Company First Sergeant, I am responsible for assigning personnel to the Company detail. This task can be quite labor intensive given the limited availability of personnel, time constraints and a full training schedule. I try to remain fair and impartial when assigning details to the ten sections within the company but still receive complaints that I choose from the same sections every month. There is no system in place to ensure random selection so it could be possible that the same sections are required to provide people to support the detail.

The details are categorized by number and rotated by company each month. The details include:

1. Fort Indiantown Gap Clearing- this detail consists of ensuring any building utilized on the training post is properly cleaned and turned-in to the Post billeting office.
2. Security- this detail consists of providing armory parking guides, gate guards, entry control point guards and security checks
3. Armory Clean-up- this detail consists of cleaning the three main buildings of the armory.
4. No detail

There was not enough information to make substantial analysis merely developing a predictor with Company assignments.

Python Code to generate random selection tool:

# Python code to demonstrate the working of

# shuffle() and uniform()

# importing "random" for random operations

import random

# Initializing list

li = ["Co\_HQ", "Space", "Protection", "Fires", "AMD","AVN","G3","G5","C2","Engineers"]

# Printing list before shuffling

print ("The list before shuffling is : ", end="")

for i in range(0, len(li)):

print (li[i], end=" ")

print("\r")

# using shuffle() to shuffle the list

random.shuffle(li)

# Printing list after shuffling

print ("The list after shuffling is : ", end="")

for i in range(0, len(li)):

print (li[i], end=" ")

print("\r")

# using uniform() to generate random floating number in range

# prints number between 5 and 10

print ("The random floating point number between 5 and 10 is : ",end="")

print (random.uniform(5,10))

**Business Question.** *What is the value added in randomizing Company duty assignments?*

There really is no value add in randomizing the selection for company details. The assigned Company Duty Roster only has three assigned areas, one issued to each of the four companies every month and a one-month break. Four assigned areas between ten sections does not require a random generator.

1. **Updated Project**

**Identifying a business need.** There has been a deteriorating workforce of full-time staff throughout the Battalion. In six months, there have been two resignations, four transfers, and three retirements. Many of these individuals held critical readiness positions within the Battalions and it has created a major backlog in Soldier Award actions and processing.

Midterm, I decided to change the project to PDF scraping to identify certain codes of Soldiers that indicate when they joined the National Guard. This code is known as ‘B1’ which ties the entry of this code into the Total Army Personnel Database- Guard (TAPDB-G) to the specific the Service Member (SM) entered the Pennsylvania Army National Guard. This information is useful in determining State and National Guard awards owed to the SM.

Company level leadership do not have access to the database to query this data and are reduced to manually searching through various pages of the SM’s Retirement Points Accounting Management (RPAM) Statement to identify the B1 code and then transposing that date to an Excel file. The RPAM statement is a service log of every duty transaction for an individual SM. The average number of Soldiers per Company in my Battalion is 130. Depending on the length of service for any given SM, the number of pages can extend to 5 or more. The current method is not only time consuming but increases the risk of human error with all the erroneous data within the RPAM statement.

Other digital platforms can be utilized such as the Defense Training Management System that provides the Pay Entry Base Date for all Soldiers; however, this cannot be utilized for Officers as they receive federal commissions and their Basic Active Service Date changes as they become officers. If the officer is prior enlisted, there is no other Open Source access to their Pay Entry Service date. This would require a query from the Battalion Human Resources Section (S1) and will be provided based on priority and current mission. Given the dwindling numbers of full-time staff, this information for State Awards would not be prioritized over Soldier retention efforts and Soldier readiness.

1. **Project Scope**

Mitigate the risk of human error and delay for State and Service awards for Army National Guard Soldiers by creating a program that identifies specific service codes and attributing effective dates, moving each to a CSV file. The types of analysis this program would provide:

* 1. Generate a by name Roster of SMs and Pay Entry Base Dates
  2. Calculate eligibility of initial and subsequent State Medals and accoutrements.
  3. Calculate Number of Soldiers eligible for Twenty-Year letters and Medals.
  4. Calculate the number of Soldiers eligible for initial and subsequent Army reserve Component Achievement Ribbons and accoutrements.

1. **Concept**

RPAM statements can be batched so you can receive the entire company’s RPAM statements in one file.

1. The Python program would scrape the RPAM statement PDF file to identify certain codes of Soldiers that indicate when they joined the Army National Guard.
2. Scrape the SM name, last four of SSN and corresponding ‘Begin Date’ of the B1 code and move to CSV File
3. Create columns in CSV file:
   1. Name
   2. Last Four
   3. Begin Date
   4. ARCAM Eligibility
   5. ARCAM Number
   6. State Medal Eligibility
   7. State Medal Number

SMs can receive more than one award as there time in service increases. After creating the CSV file, columns D-G would need to be calculated to populate the data within the columns

1. **Analysis**
2. SMs can receive more than one award as their time in service increases.
3. After creating the CSV file, the eligibility and award number fields would need to be calculated to populate the data within the columns.
4. Eligibility would be a Boolean Expression after years between ‘Begin date’ and Current date are calculated
5. The number of awards would be calculated from the PEBD Date based on the regulatory criterion for award
   1. I.e. Certain State Awards are given every three years of good service.
6. **Conclusion**

Unfortunately, I could not get the code to read the file correctly as there were several issues with how I configured my computer. Python does not work well with spaces in naming conventions. Much of this project was spent reconfiguring my administrator user folder.

The next issue was getting the PDF to scrape the text properly. Since the data was not in a true table, the code would not tie the B1 code to the correct Begin date and therefore I could not calculate eligibility. The header information of the RPAM statement was required to create a user ID which could not be found even by converting the PDF to a CSV.

I am going to continue to work through this project as I feel it will be a good portfolio piece and a useful tool for company leadership.

**Code**

#W3W School #Stackoverflow #iannaigithub

get ipython().system('pip install camelot-py')

import camelot

tables = camelot.read\_pdf('rpam.pdf')

tables

<TableList n=1>

tables.export('foo.csv', f='csv', compress=True) # json, excel, html

tables[0]

<Table shape=(7, 7)>

tables[0].parsing\_report

{

'accuracy': 99.02,

'whitespace': 12.24,

'order': 1,

'page': 1

}

tables[0].to\_csv('rpam.csv') # to\_json, to\_excel, to\_html

tables[0].df # get a pandas DataFrame!

1. Pseudonym to protect actual unit identity. [↑](#footnote-ref-1)