Contract Execution

Cheryl Bowersox

October 21, 2018

## Introduction

As part of a large IT organization one of my team’s responsibilities is to move contracts from drafting through execution, and we are frequently asked for an estimate of processing time. The current model used for estimates has a very larger window, taking anywhere between one and twelve weeks. This results in difficulty accurately forecasting project expenses and adhering to project timelines. Furthermore, delays in contract execution can result in additional expenses because other parts of a complex project are put on hold to accommodate uncertain start dates. This can also create variable deliverable quality in outcomes and risks to department goals.

The number of variables and dependencies involved in the corporate setting results in a expanded project. There are differing types of contracts, multiple reviewers and stages to the process, widely variable dollar amounts, and differing vendors. Additionally, each contract is by nature a distinct engagement, and there are external timing factors, such as year-end financial processes or quarterly forecasts, that may influence the timing.

## Other Approaches

Other approaches to the problem of contract execution timing which have been attempted have not been very successful.  
1. Amount based: For several years project managers estimated the timing based on the amount of contract, but routinely underestimated the smaller contracts.

1. Confidence Interval The current model in use examined historical data to determine an 80% confidence interval around the mean for all contracts and provides the upper bound as a pessimistic estimate. For example, all contracts greater than $500K will be executed within 60 business days. While this model is still valid, this upper bound does not provide the precise timing needed to accurately plan a larger project.

## Hypothesis for this analysis

My hypothesis is that a time-series analysis of two years of historical data, or around 400 records, could improve accuracy of predictions for execution time if all available variables are included.

By modeling the data as a time-series I hope to account for variability introduced by external cyclical factors. The inclusion variables such as type of contract, vendor, processors, amounts, and other features will refine the estimation capability. In addition to better execution predictions, any consistent patterns that emerge and analysis of outliers will enable a better understanding of key drivers for delays in execution.

## Project Goals

This project has two main objectives. The first goal is to create a valid predictive model of contract execution time that can be applied to new submissions. The second goal is to discover which variables have the greatest influence on contract timing. Understanding these variables can help make informed business decisions when developing new contract requirements.

## Data Exploration

### Data Sources

Data for this analysis was gathered from two systems and combined. The first data source was a online request intake tool. Data such as requester, vendor, intake date, and responsible analyst was gathered from this system. The second data source was a corporate contract processing system, that includes information such as contract amount, funding organization, and execution date.

Data was combined from both system for each contract submitted and executed between 2/2016 - 8/2018.

### Deidentification

Because the data sources contained confidential business data, th was deidentified to avoid disclosing any proprietary information. This include mapping suppliers and requests to an id number, scaling amounts to between 0 and 1, and similar operations for other key metrics.

### Variables in source data:



### Variable analysis

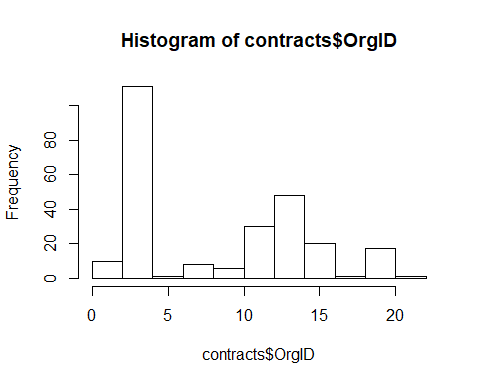
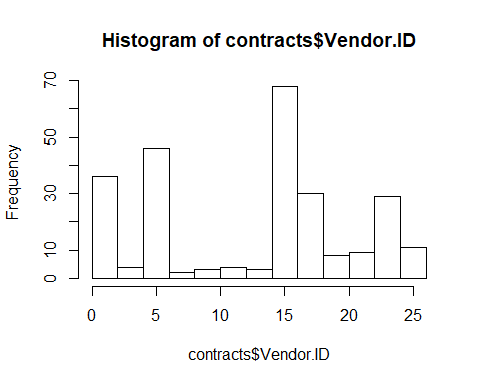
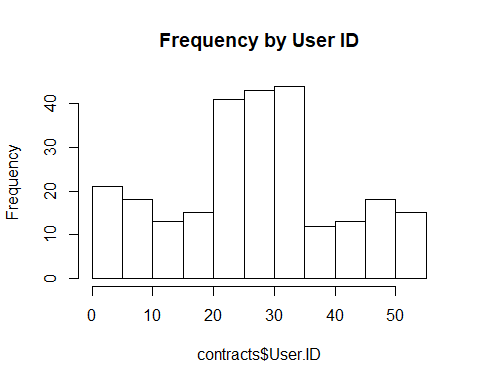
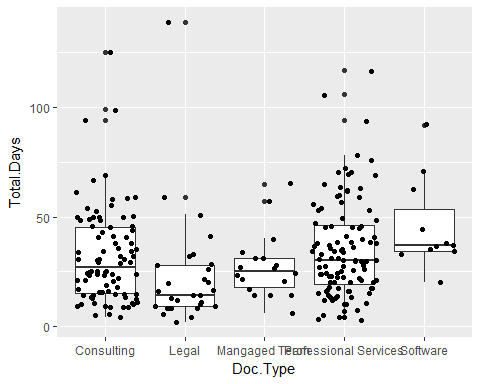
## Data Summary

The initial data set includes 253 distinct records, with 18 possible variables.

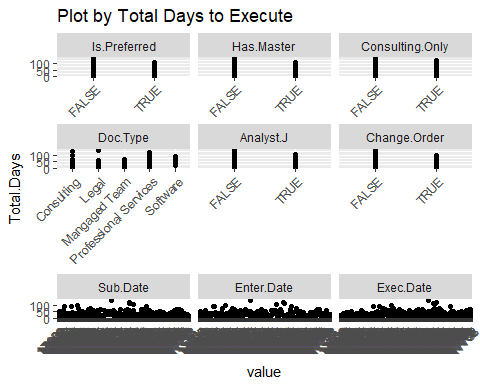
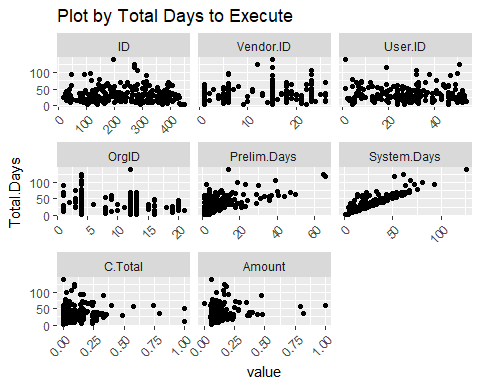
DRAFT NOTE: The number of variables will not change, however the number of contracts will change as more data will be collected for this analysis over the next 2 weeks.

The data consists of multiple categorical variables, such as document type, supplier ID, and user ID.

Because of the large number of suppliers and users, an initial analysis is done to determine if they can be grouped in any way



## [1] 74



### Discussion of grouping for users and vendors

Users who submit less than 3 contracts make up about 50% of the users. New dummy variable created New.User when the user is less than 3 submissions. The remaining users will be given dummy variables.

#make this prettier  
head(dfu,10)

## # A tibble: 10 x 2  
## User.ID count  
## <int> <int>  
## 1 24 32  
## 2 33 26  
## 3 27 15  
## 4 32 13  
## 5 29 11  
## 6 5 10  
## 7 8 8  
## 8 46 8  
## 9 26 7  
## 10 51 7

Similarly, there are 9 vendors with only 1 submission, which will be classified as a new vendor.

#make this prettier  
head(dfv,10)

## # A tibble: 10 x 2  
## Vendor.ID count  
## <int> <int>  
## 1 15 67  
## 2 1 34  
## 3 6 29  
## 4 18 28  
## 5 23 28  
## 6 5 17  
## 7 26 10  
## 8 21 6  
## 9 19 4  
## 10 20 4

There are 17 distinct possible organizations submitting contract requests. Of those 17, 8 of them have submitted less than 5 requests in the time frame in question, these are are classified together as a ‘rare’ org. The renaming organizations are modeled using dummy categorical variables.

#make this prettier  
head(dfo,10)

## # A tibble: 10 x 2  
## OrgID count  
## <int> <int>  
## 1 4 104  
## 2 13 48  
## 3 12 29  
## 4 20 13  
## 5 15 11  
## 6 1 9  
## 7 16 9  
## 8 8 8  
## 9 3 7  
## 10 9 4

The new data set with categorical variables is a sparse table with 78 columns.

may result in later groupings, but first need to determine which are influential and which are not)

# time series analysis

Clean and transformed data need to plot ts for Total Days based on the original submit date. The assumption is that the length of time a contract takes to be executed may have a time series component, and should be reviewed to rule this out.

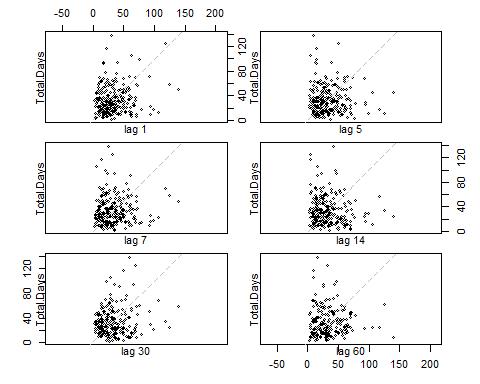
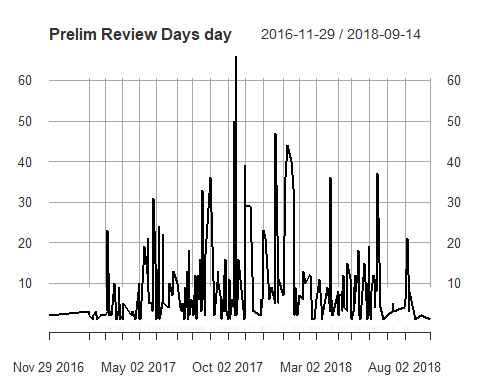
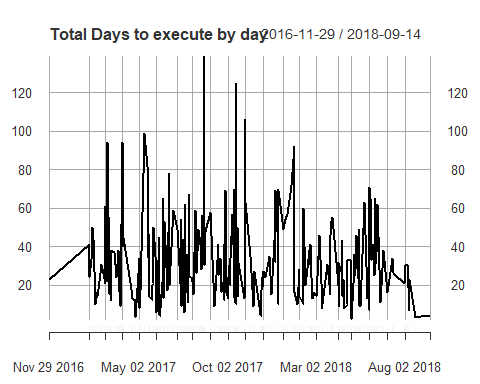
## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

##   
## Attaching package: 'xts'

## The following objects are masked from 'package:dplyr':  
##   
## first, last



## NULL

### Discussion of TS

For Total Days by submission date (Sub.Date) there is no obvious lag presented when evaluated for daily, weekly, and a monthly lag. The irregular nature of the work - some days have multiple entries, some days have no entry. Based on the preliminary plots there may be no simple signal to extract from the data as a straight time series.

## Forecasting as a TS

< Problem with the data - have not been able to solve- ts models producing error ‘R warning In get(x, envir = ns) : restarting interrupted promise evaluation’ this will need to be re-evaluated

## discussion of overall trends here

* decreased variance over time, decreased mean (?)

# Predictive models using VAR, glm, tree, random forest models

Mid-point analysis: Need further data exploration to be able to create meaningful models, glm whit 72 variables gives us an RMSE almost 0 and R = 1, which is not correct.  
Linear combination on Prelim + System = Total.Days, need to remove these values

### Models

#back to our data frame contracts4  
  
#split the data  
  
set.seed(42)   
sample = sample.int(n = nrow(contracts4), size = floor(.80\*nrow(contracts)), replace = F)  
  
contrain = contracts4[sample, ]  
contest = contracts4[-sample,]  
  
#VAR model here  
#TBD  
  
  
#To many predictors to run a GLM   
  
#GLM Modle here  
# first, start with a general linear model   
myControl = trainControl(method = 'cv', number = 5,   
 verboseIter = FALSE, savePredictions = TRUE,allowParallel = T)  
   
set.seed(143)  
  
con.m1 <- contrain %>%select(-Prelim.Days, -System.Days)  
GLM\_M1\_Data1 = train(Total.Days ~ ., data = con.m1 , metric = 'RMSE', method = 'glm',preProcess = c('center', 'scale'), trControl = myControl)

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: New.Vendors\_2

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: New.Users\_53, New.Users\_7,  
## New.Vendors\_4, New.Vendors\_2

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: New.Users\_20, New.Vendors\_10,  
## New.Vendors\_2

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: New.Vendors\_2, New.Vendors\_17

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: New.Vendors\_2

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: New.Vendors\_2

GLM\_M1\_Data1

## Generalized Linear Model   
##   
## 202 samples  
## 71 predictor  
##   
## Pre-processing: centered (71), scaled (71)   
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 162, 161, 161, 161, 163   
## Resampling results:  
##   
## RMSE Rsquared MAE   
## 1.193938 0.9969239 0.9342335

#Nueral Net Model Here  
  
  
#Random Forest Model Here

### Outstanding questions/issues:

Running into errors forecast based on ts data, possibly can move to a different model to look at the dates, or include them as an input variable.

Too many dummy variables for glm - can we group more meaningful way? does a tree model work better? can we simplify modeling by excluding user ? or vendor? based on SME research - user may be irrelevant if vendor is well known, or analyst = J. If user /vendor is removed or binned it reduces # of variables dramatically?

Do distinct models for doc types make sense? Does it make sense to model prelim vs. system separately and then combine results?

Is vendor relevant if we have vendor categories (Master/preferred, etc)?